Automatic Classification of Mammography Images Using a Convolutional Neural Network

Master Thesis

For attainment of the academic degree of
Master of Science in Engineering (MSc)

in the Master Programme Digital Healthcare
at St. Pölten University of Applied Sciences

by
Lisa Pitzl BSc
51910687

First advisor: FH-Prof. Andreas Jakl, MSc

St. Pölten, 19.05.2024

Declaration of Honour

I hereby declare that

- I have written the work at hand on my own without help from others and I have used no other resources and tools than the ones acknowledged.
- I have complied with the Standards of good scientific practice in accordance with the St. Pölten UAS' Guidelines for Scientific Work when writing this work.
- I have neither published nor submitted the work at hand to another higher education institution for assessment or in any other form as examination work.

Regarding the use of generative artificial intelligence tools such as chatbots, image generators, programming applications, paraphrasing and translation tools, I declare that

☐ no generative artificial intelligence tools were used in the course of this work.

☒ I have used generative artificial intelligence tools to proof-read this work.

☐ I have used generative artificial intelligence tools to create parts of the content of this work. I certify that I have cited the original source of any generated content. The generative artificial intelligence tools that I used are acknowledged at the respective positions in the text.

Having read and understood the St. Pölten UAS' Guidelines for Scientific Work, I am aware of the consequences of a dishonest declaration.

....

Place, Date Signature

Preface

After finishing my bachelor's degree radiology technology at the FH Gesundheitsberufe OÖ, I started my career as a radiologic technologist at the convent hospital of the Barmherzigen Brüder in Linz. Over the course of the following months, I made the personal decision to start a new academic journey, beginning with my master’s degree Digital Healthcare at the university of applied science in St. Pölten. During the second semester of my studies, I transitioned to a new position within the healthcare sector specifying in radiation oncology treatment planning, a sector of medical physics at the Ordensklinikum Linz GmbH Barmherzige Schwestern. Processing the newly attained experience and knowledge, I began to consider the potential integration of radiology technology and informatics. This was the initial inspiration for my subsequent master’s thesis.

Abstract

Introduction:

Breast cancer remains a significant global health concern, with early detection playing a crucial role in the improvement of patient outcomes. The deployment of a convolutional neural network for automated classification of mammography images has the potential to enhance the diagnostic accuracy as well as the efficacy of breast cancer diagnosis.

Methods:

This thesis addresses the necessity of developing and evaluating a convolutional neural network for classifying mammography images. The network has been developed from the ground up personally. The dataset considering mammography images of 500 patients from the radiology department of the Ordensklinikum Linz GmbH Barmherzige Schwestern was filtered and pseudonymized. The data was initially categorized according to the Breast Imaging-Reporting and Data System classification. Therefore, the models were optimized for binary and multi-class classification tasks. The performance evaluation is based on accuracy, loss, receiver operating characteristic and the confusion matrix.

Results:

The results demonstrated show that the convolutional neural network model presents promising performance in distinguishing between benign and malignant mammographic images. The validation and test accuracies were 69.49% and 70.13%, respectively. However, challenges were observed when extending the model to three and five classes. The validation accuracies varied, and the test accuracies remained relatively low. The presence of artefacts in mammography images and the variability between the Breast Imaging-Reporting and Data System classification classes were identified as key factors affecting model performance.

Conclusion:

While the models delivered satisfactory performance on training datasets, the effectiveness on unseen data was limited, indicating potential overfitting and generalization challenges. The thesis emphasizes the significance of continuous research and innovation in the development of artificial intelligence-based diagnostic tools in medicine, and especially in the radiological sector especially for the recognition and classification of mammography images.

Kurzfassung

Einleitung:

Brustkrebs ist nach wie vor ein wichtiges globales Gesundheitsproblem, wobei die Früherkennung eine entscheidende Rolle bei der Verbesserung der Behandlungsergebnisse spielt. Die Entwicklung eines neuronalen Faltungsnetzwerks zur automatischen Klassifizierung von Mammographie-Bildern hat das Potenzial, die diagnostische Genauigkeit und Wirksamkeit der Brustkrebsdiagnose zu verbessern.

Methodik:

Diese Arbeit befasst sich mit der Notwendigkeit der Entwicklung und Evaluierung eines neuronalen Faltungsnetzwerks für die Klassifizierung von Mammographie. Das Netzwerk wurde von Grund auf selbst entwickelt. Der Datensatz mit Mammographie-Bildern von 500 Patientinnen und Patienten aus der radiologischen Abteilung des Ordensklinikums Linz GmbH Barmherzige Schwestern wurde gefiltert und pseudonymisiert. Die Daten waren initial gemäß der Breast Imaging-Reporting and Data System kategorisiert. Die Modelle wurden für binäre und Mehrklassen-Klassifikationsaufgaben optimiert. Die Leistungsbewertung erfolgt anhand von Genauigkeits- und Verlustmetriken, Grenzwertoptimierungskurve und der Konfusionmatrix.

Ergebnisse:

Die Ergebnisse zeigten, dass das Modell des neuronalen Faltungsnetzwerks eine vielversprechende Leistung bei der Unterscheidung zwischen gutartigen und bösartigen Mammographie-Bildern zeigt. Die Validierungs- und Testgenauigkeit betrug 69,49 % bzw. 70,13 %. Die Erweiterung des Modells auf drei bzw. fünf Klassen brachten einige Herausforderungen mit sich. Die Validierungsgenauigkeit variierte, und die Testgenauigkeit blieb relativ niedrig. Das Vorhandensein von Artefakten in Mammographie-Bildern und die Variabilität zwischen den Klassen der Breast Imaging-Reporting and Data System Klassifizierung wurden als Schlüsselfaktoren identifiziert, die die Leistung des Modells beeinflussen.

Schlussfolgerung:

Während die Modelle in den Trainingsdatensätzen eine zufriedenstellende Leistung zeigten, war die Effektivität bei den ungesehenen Daten begrenzt, was auf mögliche Überanpassungen und Generalisierungsprobleme hinweist. Die

Arbeit unterstreicht die Bedeutung kontinuierlicher Forschung und Innovation bei der Entwicklung von auf künstlicher Intelligenz basierenden Diagnosewerkzeugen in der Medizin und insbesondere im radiologischen Bereich für die Erkennung und Klassifizierung von Mammographie-Bildern.

Table of Content

1 Introduction

Breast cancer is a significant health burden worldwide. Mammography screening programs play a crucial role in early detection, highlighting the need for accurate and efficient diagnostic tools such as Convolutional Neural Networks (CNN). This thesis aims to develop and evaluate a CNN-based solution for the automatic classification of mammography images.

1.1 Background of the Work

According to the global cancer statistics of 2018 that covered 36 types of cancer in 185 countries, breast cancer is the second leading cause of cancer death and the most commonly diagnosed cancer among women worldwide (Bray et al., 2018).

As reported by Wild et al. (2020) the International Agency for Research on Cancer (IACR) of the World Health Organization (WHO) outlined that lung cancer is the most common cancer leading to death in both men and women. However, looking at the genders individually, prostate cancer is the most frequent cancer in men and breast cancer in women.

The global incidence of new breast cancer among women is approximately 2.3 million per year, which represents 11.7% of all cancer incidents worldwide (Sung et al., 2021).

Invasive breast cancer will manifest in around 13% of all women in America, which means one in eight American women is confronted with the diagnosis breast cancer at some point in their life (Abdelrahman et al., 2021).

According to the breast cancer screening program “früh erkennen” by the Österreichische Gesundheitskasse (2023) about 5,000 women in Austria are diagnosed with breast cancer every year, which corresponds to one in eight women in this country. Organized mammography screening programs are intended to counteract the mortality rate from breast cancer. The goals of the organized screening programs are to reduce the mortality rate of breast cancer in

the long term and to guarantee a high quality of mammography. In general, this screening examination is carried out to detect women who still have asymptomatic breast cancer and to treat them successfully.

Early identification of the carcinoma increase the likelihood of survival and decrease the cost of therapy (Shan et al., 2023).

As stated by Mohi ud din et al. (2022) mammography is the gold standard and the popularly employed method for diagnosing breast cancer at an early stage of the disease and in women with no signs of the disease. The X-ray images of the breast are called mammograms.

The development and implementation of a CNN for the automated classification of mammography images is a highly compelling and active research area. It addresses the need to enhance breast cancer diagnosis precision and effectiveness, which is a crucial component of personalized healthcare and healthcare systems globally. In accordance with the use of artificial intelligence in healthcare and the digital transformation of medical practices, this research examines the intersections between data science, technology, and healthcare.

1.2 Objectives

The purpose of this thesis is to develop a CNN for the automated classification of mammography images. The main topic is image analysis, with a particular focus on the detection of breast cancer through the analysis of mammography images.

The goal is developing and evaluating a CNN-based solution for the automated classification of mammography images and achieving high levels of accuracy.

A dataset of 1,480 pseudonymized mammography screening examinations with a total of 500 individual patients was extracted from the radiology department of the Ordensklinikum Linz GmbH Barmherzige Schwestern. The examinations were performed in the period 2021–2023. The main contribution of this thesis is the development and description of the process to train and run a CNN-based method for classifying mammography images into two, three or five classes. Whereas the five classes are categorized according to the Breast Imaging-Reporting and Data System ® (BI-RADS®) classification.

Those who can benefit from this work include patients, healthcare professionals and researchers.

1.3 Research Questions

Research question 1: How can a CNN be developed and evaluated?

Research question 2: How can a CNN, with automated classification of mammography images, help improve breast cancer diagnoses?

1.4 Methodical Approach

The methodology is crucial in developing a neural network for medical image classification. It involves systematic procedures for data collection, preparation, model setup, architecture design, training, testing, and prediction. The thesis employs both qualitative and quantitative approaches. A comprehensive literature review, as a qualitative method, provides a foundation for the information presented in this thesis and the development of the model. The research includes various academic databases and scientific papers to gain insights into image analysis, breast cancer diagnosis, and CNN-based classification techniques. The quantitative approach involved the evaluation of the model's performance using a number of metrics, including accuracy, loss, the Receiver Operating Characteristic (ROC) curve, and the confusion matrix.

2 Background and Related Work

The research was motivated by the urgent issue of breast cancer, which is as mentioned in the research conducted by Mohi ud din et al. (2022) the second most common cancer worldwide and the second leading cause of death for women. The three main categories of breast carcinoma are: ductal carcinoma, invasive carcinoma and invasive lobular carcinoma. The ductal carcinoma originates in the milk ducts and has not yet grown into the surrounding tissue. The invasive carcinoma, which is the most common type of mamma carcinoma, invades in the surrounding tissue. The invasive lobular carcinoma develops from the glands of the breast and is considered a form of invasive carcinoma.

As explained by Deutsches Krebsforschungszentrum (2023) a tumor develops if cells of the mammary gland – which produces milk for young offspring – begin to grow uncontrollably. If this tumor subsequently grows invasively, damages and destroys the surrounding tissue it is referred to as malignant breast cancer.

Wessel & Wyant (2021) of the American Cancer Society outline that the stage of breast carcinoma indicates how far the malignant cancer cells have spread from the original tumor and invaded healthy tissue. The tumor first invades nearby healthy tissue and then lymph nodes – in the case of breast cancer, the axillary lymph nodes – from where the cancer cells can spread to other parts of the body.

According to Isosalo et al. (2023) screening mammography is an effective method for detecting breast cancer. During screening mammography, women without symptoms are imaged in order to detect malignant findings at an early stage of the disease. Each breast is examined individually using low-energy X-rays. Two projection images are usually taken of both breasts. The projections are bilateral craniocaudal (CC), as shown in figure 1 on the left side, and mediolateral oblique (MLO) images, shown in figure 1 on the right side. Afterwards a radiologist visually interprets these images. It is recommended to employ both the CC and MLO projection for detecting abnormalities, as a true anomaly is typically detectable in two sperate views (Abdelhafiz et al., 2019).

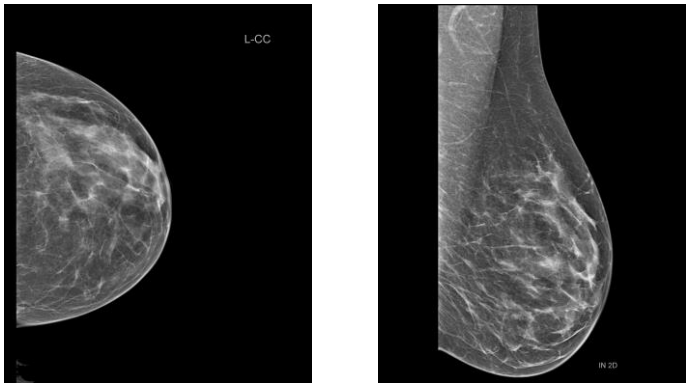

Figure 1 Mammograms CC (left) and MLO (right). Ordensklinikum Linz GmbH Barmherzige Schwestern, 2023.

Mammography is very sensitive regarding calcifications, microcalcifications or a cluster of calcifications and achieves very good results in their detection as mentioned by Mohi ud din et al. (2022). Therefore, mammography is commonly used to diagnose breast cancer in women without symptoms. The technical aspects of mammography also make it suitable for screening and routine screening applications. Therefore, mammography is seen as the golden standard for diagnosing breast cancer at an early stage of the disease. Further imaging modalities for the detection of the mamma carcinoma are ultrasound, histopathology, Magnetic Resonance Imaging (MRI), Computer Tomography (CT) and Positron Emission Tomography (PET).

This is why mammography is regarded as one of the most prevalent procedures for the early detection of breast cancer, with a significant impact on the mortality rate, as it detects cancer at the earliest stage of the disease (Abdelhafiz et al., 2019).

Due to the increasing amount of mammography images and data generated by widespread screening programs, radiologists are often challenged to perform accurate assessments within a reasonable time frame (Ragab et al., 2021).

Isosalo et al. (2023) point out that the interpretation of mammography images is influenced by several factors that make the detection of tumors difficult. Dense breast tissue is one of these factors. In addition, the structures to be imaged - in the case of mammography, the breast - vary greatly in size and shape. Some abnormalities can be less than one millimeter in size. It is this aspect that demands very high quality and precision in image resolution. Comprehensively, the diagnosis requires a high level of experience and training.

2.1 BI-RADS®

The interpretation of mammography images in nearly all examinations is classified according to the BI-RADS® classification. It should be noted that all final scores in the BI-RADS® classification are based on a thorough assessment of the relevant mammographic features according to the American College of Radiology et al. (2013). Table 1 provides an overview of the BI-RADS® assessment categories.

BI-RADS® Assessment Categories		
Category 0	Incomplete – Need additional imaging evaluation and/or prior mammograms for comparison	
Category 1	Negative	
Category 2	Benign	
Category 3	Probably benign	
Category 4	Suspicious	Category 4A: Low suspicion for malignancy
		Category 4B: Moderate suspicion for malignancy
		Category 4C: High suspicion for malignancy
Category 5	Highly suggestive of malignancy	
Category 6	Known biopsy-proven malignancy	

Table 1 BI-RADS® Assessment Categories. Own illustration based on American College of Radiology et al. (2013).

Only mammography images from BI-RADS® categories one to five were used for this thesis.

2.1.1 Category zero – Incomplete

In category zero, the examination is incomplete and additional imaging and/or previous mammograms are required for comparison following the guidelines of American College of Radiology et al. (2013). This category is used in screening situations or in diagnostic mammography reports when, for example, the equipment or personnel to perform the required diagnostic procedures are not immediately accessible or the patient cannot or will not wait for the entire diagnostic examination to be completed. The usage of category zero should be avoided in diagnosing mammograms necessitating further assessment by MRI.

For this purpose, the radiologist should provide a final assessment before the MRI scan is executed. Ultimately, category zero is used when the mammographic assessment is incomplete in contrast to the other BI-RADS® categories, where the mammographic assessment is complete.

2.1.2 Category one – Negative

Category one describes a normal mammogram performed as part of a routine mammography screening (American College of Radiology et al., 2013).

2.1.3 Category two – Benign

According to American College of Radiology et al. (2013) category two is a normal mammogram, similar to category one, but the radiologist decides to classify the finding as benign.

The breast in the mammogram has a characteristic benign appearance and can be confidently described as benign. This includes calcified fibroadenomas, skin calcifications, metallic foreign bodies (e.g. core biopsies and surgical clips) and fat-containing lesions (e.g. oil cysts, lipomas, galactoceles and mixed density hamartomas). In addition, intramammary lymph nodes, vascular calcifications, implants or architectural distortions may be described, but these do not in any way indicate malignancy.

If the report does not describe these structures, the mammogram could also be classified as category one. Category one and category two findings both suggest the absence of malignancy in the mammographic result. The difference is that reports of category two mention one or more particular benign mammographic findings, while category one is assigned when such findings are not specified, even if they might be present.

2.1.4 Category three – Probably Bening

As described by American College of Radiology et al. (2013) category three findings have a probability of malignancy of less than or equal to 2%, but greater than a benign finding with a 0% probability of malignancy.

Various prospective clinical trials have indicated the safety and effectiveness of regular mammographic surveillance rather than biopsy for certain mammographic findings. For the majority of likely benign findings, initial surveillance is performed at shorter intervals of six months until long-term stability of two to three years can be proven. In some cases, however, a biopsy may be necessary.

2.1.5 Category four – Suspicious

Category four findings do not contain the typical presentation of malignancy, but they present enough concordance to medically justify a recommendation for a biopsy as mentioned by American College of Radiology et al. (2013). Consequently, the majority of recommendations for breast interventional procedures are based on category four findings.

The upper limit of the probability of malignancy for the assessment of category three is 2%, while the lower limit for the assessment of category five is a probability of malignancy of 95%. The Category four assessment encompasses the wide range of malignancy probability in between.

Category four is divided into three subgroups, designated A, B, and C. Category four A describes findings with low suspicion for malignancy, which corresponds to a likelihood of malignancy of > 2% to ≤ 10%. Category four B describes findings with moderate suspicion for malignancy, which corresponds to a likelihood of malignancy of > 10% to ≤ 50%. Category four C describes findings with high suspicion for malignancy, which corresponds to a likelihood of malignancy of > 50% to < 95%.

In this thesis, however, no subcategories were differentiated within category four.

2.1.6 Category five – Highly Suggestive of Malignancy

Category five assessments are associated with a very high probability of malignancy of ≥ 95% as outlined by American College of Radiology et al. (2013). The current standard of oncology treatment involves the tissue diagnosis of malignancy using percutaneous tissue sampling to facilitate treatment options.

2.1.7 Category six – Known Biopsy-Proven Malignancy

According to American College of Radiology et al. (2013) category six include mammography examinations conducted after biopsy evidence of malignancy. This encompasses mammography images obtained following percutaneous biopsy but before complete surgical removal. Nevertheless, no additional mammographic abnormalities were identified in the mammography, with the exception of the previously diagnosed cancer, which may necessitate further assessment.

2.2 Radiology and Artificial Intelligence

According to Ragab et al. (2021) computer aided detection systems (CAD) supported by artificial intelligence offer great potential for application in various sectors of healthcare. The development of CAD systems for radiological image data has been going on for several decades. The diagnostic process has been made easier with the help of CAD systems, and it can improve diagnostic accuracy. Particularly in the area of screening programs, such as the evaluation of breast cancer using mammography, CAD systems simplify the visual interpretation. Furthermore, the outcome can be considered as a second opinion.

Regarding the second research question of this thesis, a CNN that automatically classifies mammography images can improve diagnostic accuracy or simplify visual interpretation. The radiologist would no longer have to examine every mammogram in detail but would receive a pre-selected selection of critical image data from the CAD systems, thus have a better overview of the number of findings from the many screening examinations.

Currently, the technology for analyzing and evaluating images is changing from traditional artificial intelligence approaches to deep learning systems and particularly those that use CNNs (Isosalo et al., 2023).

As outlined by Abdelhafiz et al. (2019) the application of CNNs in medical imaging has the potential to enhance the accuracy of radiological diagnoses. By providing a precise quantitative analysis of suspicious lesions, CNNs can assist radiologists in making more informed decisions. Current research has demonstrated that the utilization of deep learning methodologies has the potential to reduce the incidence of human error in the diagnosis of breast cancer by 85%. The recent CNN models have been developed to enhance the capacity of radiologists to identify even the most understated breast cancers at an early stage, suggesting that further analysis should be undertaken. Recent research has indicated that CNNs have been employed to describe lesions with a standardized generated description, which may assist radiologist in achieving greater accuracy in decision-making. Additionally, CNNs are employed in mammography for the purpose of lesion detection and localization, image retrieval, classification, and risk assessment.

Deep learning is used in a variety of medical applications as outlined by Mohi ud din et al. (2022). The technique of deep learning has already achieved excellent results in retrospective studies, in the biomedical sector and industry especially. However, in order to continue this technological progress, prospective studies and external validation are required to move from the research setting into clinical practice and clinical decision-making aid.

2.3 Machine Learning

According to Müller & Guido (2016), machine learning is a research domain at the crossing point of multiple specializations like statistics, artificial intelligence as well as computer science. Other terms for this specific branch of knowledge are predictive analytics or statistical learning. The core process is extracting knowledge from data. The flowchart presented in figure 2 illustrates the distinction between supervised learning and unsupervised learning in the context of machine learning (Bunker & Thabtah, 2019).

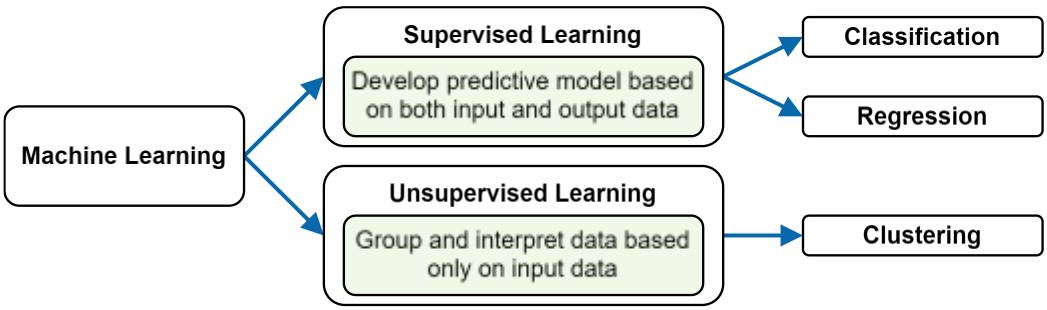

Figure 2 Distinction between supervised learning and unsupervised learning. Own illustration based on Bunker & Thabtah, 2019, p.28.

2.3.1 Supervised Learning

Müller & Guido (2016) pointed out that the most popular and successful algorithms in the machine learning sector are those which generalize known examples and use them to automate decision-making processes. Therefore, the task is to predict an outcome from a given input in order to be later on able to make a reliable and precise prediction for new data the model has never seen before. This specification of machine learning is called supervised learning: Corresponding pairs of inputs and desired outputs are made available for the algorithm which subsequently discovers a method to generate the intended result based on the input. Even though the process of creating datasets might be tedious, the supervised learning algorithm often has a performance that is simple to measure and well understood. Supervised learning algorithms are one of the most successful and commonly used types of machine learning.

However, as outlined by Müller & Guido (2016) there are two types of challenges in the research field of supervised learning: classification and regression. In short, the task of regression is to predict values, whereas classification predicts classes (Géron, 2019).

Classification

For the classification problem, the objective is to predict a class label from a predefined list of options as mentioned by Müller & Guido (2016). Binary classification distinguishes between exactly two classes, while in the multiclass classification the model has to distinguish between more than two classes. In the following task of classifying mammograms in the corresponding BI-RADS® classifications the problem is a multiclass classification one.

Regression

According to Müller & Guido (2016) the objective for the regression problem is to predict a continuous number or in programming terms a floating-point number within a given range, such as predicting a person's annual income based on their education, age, and location. By considering whether there is a continuity in the output, or not, the differentiation between a classification and regression task can be made. If there is a continuity in the output – think of any number in a given range – then the problem is a regression task.

2.3.2 Unsupervised Learning

Unsupervised learning is another type of machine learning mentioned by Müller & Guido (2016). In contrast to the supervised learning method, the unsupervised learning algorithm is only provided with known input data, but no output data. Therefore, the task for the learning algorithm is to extract knowledge from the given data. The crucial point here is to develop an algorithm that can generate knowledge or a pattern from the data based solely on the input data. Although there are successful examples of those algorithms, they are often more difficult to understand and evaluate.

2.4 Neural Network

Sarker (2021) describes deep learning as a sub-area of machine learning and due to the ability to process large amounts of data, there is a trend towards using this capability in many areas.

In recent years a group of algorithms – known as neural networks – have gained popularity in the machine learning domain under the name deep learning as noted by Müller & Guido (2016). Due to their ability to capture information from a large amount of data and build complex models, neural networks have become state-of-the-art models in many machine learning applications. To describe a neural

network, one can use a method called multilayer perceptron (MLP), which can form the basis for more complex deep learning methods. MLPs can be used for both classification and regression problems and are also referred to as (vanilla) feed-forward neural networks or simply neural networks. MLPs undergo multiple phases of processing to reach a decision and can be considered as generalizations of linear models, which use a linear function of the input features to create a prediction.

The formula for the prediction looks as follows:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \ldots + w[p] * x[p] + b$$

- $\hat{y}$ weighted sum of the inputs,
- $x[0]$ to $x[p]$ input features,
- $w[0]$ to $w[p]$ learned coefficients (Müller & Guido, 2016).

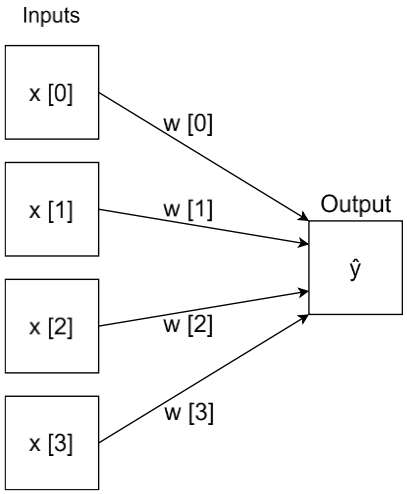

Figure 3 Visualisation of the formula for the prediction, in which the inputs and outputs are represented as squares and the coefficients are the connections between the squares. Own illustration based on Müller & Guido, 2016, p. 105.

In figure 3, based on a visualization of Müller & Guido (2016), the nodes on the left side illustrate the input features, while the node on the right side displays the output. The learned coefficients are represented by the connecting lines. The output is therefore a weighted sum of the inputs. This representation reflects the operation of a single perceptron. However, it is essential to note that within a MLP this process is repeated multiple times, incorporating hidden units that represent intermediate processing steps between the input and output layers. The incorporation of these hidden units results in additional layers of computation, thereby enhancing the capability of the network to capture complex relationships within the data. The number of nodes in the hidden layers can be set by the user

individually allowing for fine-tuning of the architecture in the network to accomplish specific tasks. If there are more layers correlating there are more coefficients – also known as weights – and the complexity of the network increases. Summarizing, two aspects that affect the complexity of the neural network can be defined individually: the number of layers, especially hidden layers, and the number of units in each hidden layer. To enhance the power of the model beyond that of a linear model an additional step has to be made as computing a series of weighted sums is mathematically equivalent to computing just one weighted sum. Therefore, a nonlinear function is implemented to the outcome after the calculation of a weighted sum for every hidden unit. Typically, the rectifying is used, although a more detailed description can be found in chapter 3.4.2. The output of this function is used in the sum that calculates the result $\hat{y}$.

2.5 Convolutional Neural Network

In the field of analyzing and classifying images – and especially medical images – the CNN, as a member of the family of artificial neural networks (ANN), is the most common model in the field of deep learning (Mohi ud din et al., 2022). Since 2012, CNNs have grown in popularity and attention, driven by advances in computing power, the accessibility of more affordable hardware, open-source algorithms, and the increase of big data (Abdelhafiz et al., 2019).

The CNN is a unique type of multilayer neural network among the various deep learning architectures (Ghosh et al., 2020). According to Wuttke (n.d.) the difference between a classic neural network and a CNN is to be found in the architecture of the networks. In a CNN, the hidden layer is based on a sequence of convolution and pooling operations. During the convolution layer a so-called kernel is pushed over the data and a convolution is calculated in the process. Abdelhafiz et al. (2019) states that the term "deep" typically refers to the depth of hidden layers within a CNN.

Deep learning efficiently analyses correlated data and can therefore automatically select features from it as outlined by Mohi ud din et al. (2022). With a sufficient amount of training data, a CNN is capable to extract complicated and multi-layered hierarchical features from images. With the algorithm of a CNN high-level features can directly be extracted from the image data and deliver faster and more accurate results than classical machine learning algorithms. With this end-to-end processing method, the algorithm is able to learn features itself. The image data is used to learn hierarchical features. This technology is primarily used in image classification, computer vision, face recognition, and the processing of audio or

video material. However, the CNN technology also shows very good results in medical image analysis and categorization. The advantages of the CNN architecture include the shared usage of parameters and the sparse connections. Relevant features are efficiently extracted from image pixels through the use of filters.

Abdelhafiz et al. (2019) indicated that CNNs has been demonstrated to be particularly effective in context of image analysis tasks. Similarly, Isosalo et al. (2023) reported the effectiveness of CNNs in the detection and diagnosis of cancer with encouraging the outcomes. The CNNs have been used to detect and classify cancer based on various radiological images (mammography, CT, ultrasound, MRI). In the paradigm of CNN, relevant representations are learned directly from the data.

The approaches used in previous studies to classify lesions have been discussed by Abdelhafiz et al. (2019). The authors discuss approaches that focus on categorizing mammographic images into two categories. The objective is to develop a CNN to assess the likelihood of images being normal, containing masses and/or microcalcifications. In other studies, CNNs are typically employed to categorize mammographic images into benign or malignant classes, or even three categories, including the absence of tumors. Furthermore, some studies have focused on the progression of malignancy of masses and the detection of microcalcifications on mammography. Furthermore, CNNs have demonstrated considerable potential for the development of a novel, short-term risk prediction approach with enhanced performance in the detection of early, abnormal symptoms within negative mammography images. Breast density is regarded as a considerable predictor of breast cancer risk.

According to Wang et al. (2017) breast arterial calcifications identified on mammograms may function as a valuable indicator of cancer risk factors. The results of this study demonstrated that the CNN exhibited a comparable level of recognition to that of human experts.

The study conducted by Carneiro et al. (2015) addressed the categorization of masses through the utilization of a pre-trained multi-view CNN. The approach involves classifying an entire mammogram by identifying characteristics from each breast perspective, training distinct CNNs for each perspective, and merge the features into an integrated CNN model. Afterwards the model generates a prediction, assessing the probability of the patient developing breast cancer.

3 Methodology

The methodology employed in the development of a neural network for the classification of medical image has a significant impact on the robustness, reliability, and efficiency of the model. This chapter describes the systematic approach used to collect and prepare the data as well as the setup, development, training, testing, and prediction of the neural network.

The methodology includes both qualitative and quantitative methods to develop a model for image analysis while utilizing established academic principles at the current state of research.

One qualitative method used in this thesis was a comprehensive literature research. This provided insights into the theoretical foundations, methodological approaches, and best practice examples in the field of image classification. The development of the model in this thesis is based on the findings gathered in the literature research. To conduct a comprehensive literature search radiology textbooks, online books, articles, and journals from various databases were used. The following literature databases can be mentioned as examples: ScienceDirect Elsevier, SpringerLink Portal, Thieme eJournals, Google Scholar, PubMed Central, the ACM Digital Library, Institute of Electrical and Electronics Engineers (IEEE) Xplore®. The literature research is critical to gain a comprehensive understanding of existing research in medical image analysis, breast cancer diagnosis, and CNN-based image classification. Identifying the most recent developments, cutting-edge techniques, and best practices in these areas is important.

This thesis employed quantitative methods to assess the performance of the neural network. The metric employed was the accuracy, which provides a numerical measure of the classification performance. Furthermore, the loss function, the ROC curve and the confusion matrix were used.

Throughout this chapter of the methodology, the processes involved in the individual phases up to the neural network will be presented. The overall aim is to provide a comprehensive understanding of the methodology used, based on both theoretical principles and practical considerations. The methodology's key components include data collection, data preparation, setup, model architecture, training, and testing.

3.1 Data Collection

The data collection chapter explains the selection of the dataset and the process of data collection.

According to (Abdelhafiz et al., 2019), training a CNN model with an non-annotated dataset represents a significant challenge for current research. In a non-annotated dataset, the input images of the model are labeled such as regular or cancerous, without specifying the precise location of the anomalies. The training of CNNs for the classification of non-annotated datasets remains an open area of research. This thesis addresses the research area of the classification of non-annotated datasets by utilizing one whose image data has only been labeled to a specific class.

3.1.1 Dataset

From the radiology department of the Ordensklinikum Linz GmbH Barmherzige Schwestern pseudonymized mammography images are used as data. Prior to initiating the data extraction, a permit had been obtained from the radiology department for the use of the existing data for the master thesis. A dataset compromising 1,480 mammography screening examinations (with 500 unique patients) conducted over the period 2021–2023 were extracted. The extracted data originated from a data source called the picture archiving and communication system (PACS). To extract the relevant image data from the PACS, the radiology department provided a list of patient data (name, date of birth, date of examination, BI-RADS® classification), sorted according to the BI-RADS® classifications. Using this list, the relevant mammography images could be searched for in the PACS and exported pseudonymised. The selection of the image data was only possible with the help of the patient data provided by the radiology department. As soon as the image data had been extracted, the pseudonymised export of the image data was carried out in the PACS.

The digital mammograms in the dataset originate from the PACS in Digital Imaging and Communication in Medicine (DICOM) format.

As described by Weissensteiner (2012) the DICOM standard is a multivendor-capability communication standard that specializes in the exchange of radiological information. The DICOM standard is intended to enable imaging and image-processing devices to exchange images and the associated information, like name, date etc. with each other. Image material from different modalities can be merged and therefore enhance the communication of diagnostic imaging information.

According to the DICOM standard each imaging data from an examination was coupled with textual information. Among the data were for example, patient age, patient name, or the examination date. However, the entire image data was pseudonymised for this master thesis. The actual pseudonymisation process took place during the export of the data from the PACS to the image data platform. Figure 4 shows the excerpt from the PACS where the “DICOM file export (anonymous)” is selected under the target tab to carry out the pseudonymised export of the selected image data. The pseudonymised data could then be exported from the storage area in the image data platform. The text information and all metadata of the DICOM data was removed. After the pseudonymisation process, the files are given a file name consisting of two number combinations followed by “Mammo links”, “Mammo rechts”, or “Mammo beidseits”, that indicates which breast side has been imaged. The first number combination is the patient identification (ID), which is assigned to each patient within a hospital. However, this patient ID can only be traced back to a patient with internal access to the specific hospital system. The second and third number combinations in the file name are the date and time of the examination. Due to the fact that the files are exported from the PACS with this file name, which contains the patient ID, the data is pseudonymised and not anonymised. These pseudonymized mammography images in DICOM format are classified according to the BI-RADS® classification. In order to know the BI-RADS® classification of the exported image data, the export was carried out sequentially. Firstly, all mammography images of patients belonging to the BI-RADS® one classification, for example, were exported. Once all files belonging to the aforementioned BI-RADS® classification had been exported, they were subsequently transferred to a folder with a file name corresponding to the relevant BI-RADS® class, for instance, the file name “BIRADS1”.

Figure 4 Excerpt from the PACS for performing the pseudonymised export. Own illustration.

In order to structure the extracted dataset, an exclusion was performed according to previously defined restrictions. The dataset was structured in a manner that excluded examinations in which the two standard views per breast – namely MLO and CC – were not present.

It should be noted that all mammography examinations were integrated in the dataset, including those with suspected malignant disease, i.e. with a referral for biopsy, an already known carcinoma, or a breast operation.

Regarding the imaging devices, the digital mammograms in the dataset originate from two mammography devices from different manufacturers. One is the “MAMMOMAT Revelation” mammography device from Siemens Healthcare Diagnostics GmbH. And the second device is the “Selenia® Dimensions® Mammography System” from Hologic, Inc.

3.1.2 Division of the Dataset

The entire dataset consists of 1,480 mammography images. Figure 5 illustrates the distribution of the image data across the BI-RADS® classes. Since the amount of data is of crucial importance for this thesis and the resulting model, it was consciously decided not to achieve an exact balance of image data in the different BI-RADS® classes, because otherwise image data would need to be removed to equalize the number of images across the classes.

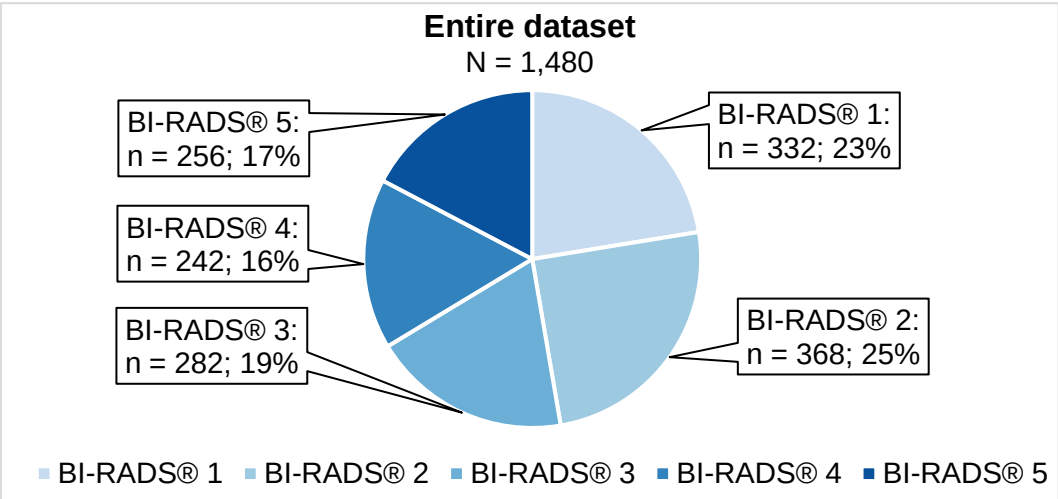

Figure 5 Division of the entire dataset according to the BI-RADS® classes. Own illustration.

As noted by Müller & Guido (2016) in order to evaluate the performance of the model, new data which the model has not seen before and for which labels have been defined is presented to it. Therefore, the gathered labelled data is divided into two parts. The training data or training set is used to construct the machine learning model, while the remaining data is used to evaluate the model's effectiveness, and is referred to as the test data, test set, or hold-out set. The split into training and test datasets is done to measure how the model generalises to new, previously unseen data. It is crucial to note that the focus is not on the model's

performance on the training dataset, but rather on its ability to make accurate predictions for data that was not observed during training.

The aim was to divide the dataset into an 80% training set and a 20% test set. However, to avoid potential biases in model accuracy, the mammography images of one breast, each with the MLO and CC projections, were kept together in one dataset. This ensures that the same breast with different projections is not included in two different datasets, such as training and testing. This leads to 79.86% in the training dataset, which are 1,182 images and 20.14% in the test dataset, which contain 298 images. In the figure 6 the division in train and test dataset is visualized.

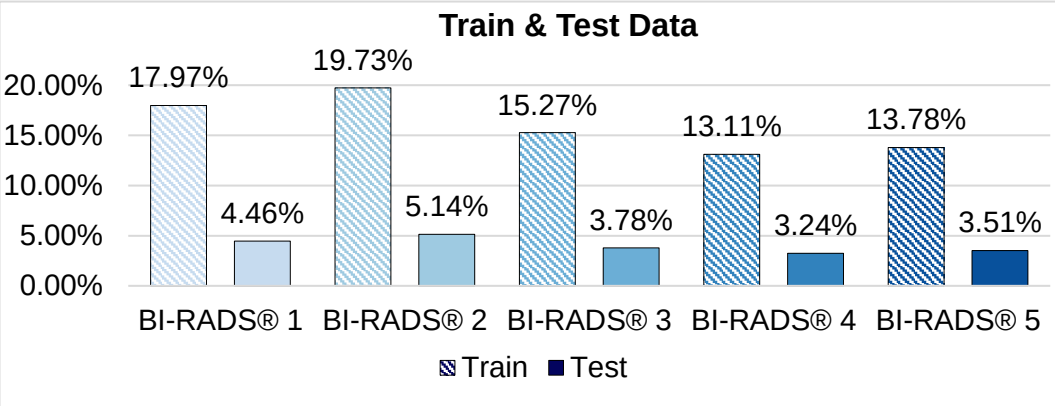

Figure 6 Division in train and test data. Own illustration.

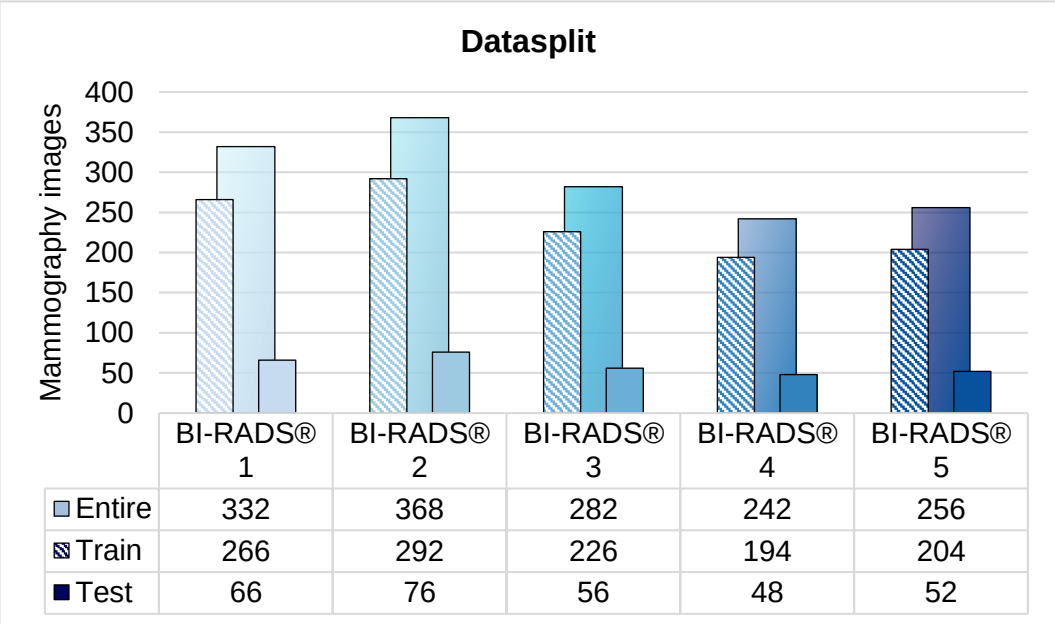

Figure 7 Datasplit of the entire dataset. Own illustration.

Figure 7 illustrates the division of the data into training and test sets, as well as an overview of the entire dataset. This visualization provides a comprehensive understanding of the composition and division of the dataset, which is advantageous for the development of the neural network.

This division of the data into training and test datasets was done manually and locally in the file explorer. This directory structure is also visualized in figure 8. There is a folder called `Mammograms_PNG_Split_train-test` which contains the `train` folder and the `test` folder. Within these two folders there are five subfolders with the respective BI-RADS® classes: `BIRADS1`, `BIRADS2`, `BIRADS3`, `BIRADS4`, `BIRADS5`.

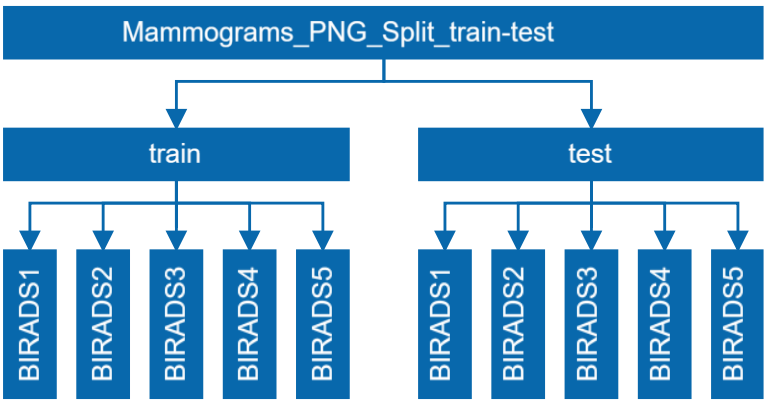

Figure 8 Directory structure in the file explorer. Own illustration.

The dataset under consideration has been categorized according to the BI-RADS® classification one to five. In order to develop a CNN model, the data was initially summarized and divided into two and then three classes. Figure 9 illustrates the division in different two, three or five classes. Divided into two classes, the first class includes BI-RADS® classifications one, two and three, while the second includes BI-RADS® classifications four and five. If the dataset was divided into three classes, the normal class contains the BI-RADS® classifications one and two, the benign class contains the BI-RADS® classification three, and the malignant class contains the BI-RADS® classifications four and five.

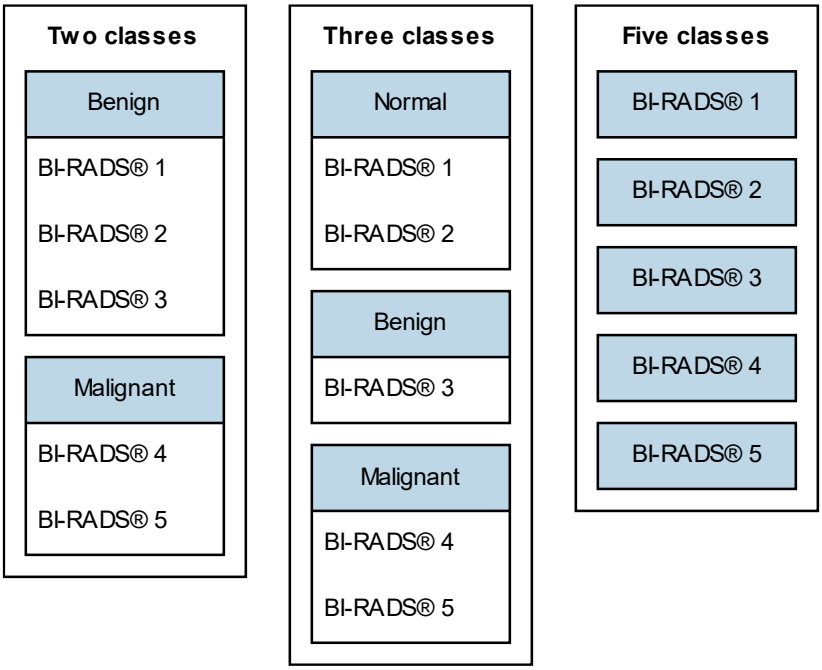

Figure 9 Data categorization and class division. Own illustration.

3.2 Setup

The following chapter covers the software tools used for the model development. Google Colab was used to develop this CNN. By providing free access to the GPU, Colab provides a hosted Jupyter Notebook service that facilitates model training and experimentation without requiring any setup.

3.2.1 Python

Müller & Guido (2016) state that the lingua franca for machine learning tasks has become Python. Python has a great functionality for general and special purposes and is used for many data science applications. It is a combination of a programming language for general objectives and the ease of use of specific scripting language for this domain. Due to the fact that machine learning as well as data analysis are iterative processes, the major advantage of Python is the possibility to interact directly with the code when using tools like the Jupyter Notebook. For these processes it is crucial to have tools that enable rapid iteration and seamless interaction.

Jupyter Notebook is an interactive environment to execute code in a browser, a useful tool for preliminary data analysis and besides that it facilitates the integration of text, code and graphics (Müller & Guido, 2016).

3.2.2 Libraries and tools

To start a CNN, several libraries and modules need to be imported. The most essential libraries required for the machine learning task are outlined below.

Matplotlib

For making high-quality visualisations Müller & Guido (2016) mentioned matplotlib as the primary scientific plotting library. Matplotlib offers various visualisation functions, including histograms, line charts, scatter plots and more. The visualization of the data can provide valuable insights.

NumPy

NumPy is used for scientific computing and therefore a foundational package in Python including high-level mathematical functions as well as the functionality for multidimensional arrays (Müller & Guido, 2016).

Python Imaging Library

In order to work with the PNG data, the Pillow library was installed. According to Clark (2024) Pillow is a user-friendly-version of the Python Imaging Library (PIL). The PIL enhances the image processing capabilities of the python executor by offering support for a wide range of file formats, efficient internal rendering, and powerful image processing capabilities.

TensorFlow & Keras

According to Abdelhafiz et al. (2019) TensorFlow was developed by the Google Brain team to support machine learning applications and was released as an open-source in 2015. It is a Python-based library for distributed numerical calculations that enables efficient training and execution of large neural networks by spreading computations among potentially hundreds of servers equipped with multiple high-performance graphics processing units (GPUs). This can speed up computation, making it essential for deep learning methods with large datasets.

For this project TensorFlow version 2.15.0 was used.

As explained by Géron (2019) Keras is a high-level application programming interface (API) for deep learning that simplifies the training and execution of neural networks. Known as `tensorflow.keras`, TensorFlow has its own implementation

of Keras, which supports advanced TensorFlow features such as efficient data loading.

Ramasubramanian & Singh (2019) mentioned that Keras and TensorFlow are indeed popular and widely used open-source deep learning libraries for Python users. Their flexible interfaces enable creating powerful neural networks and keeping up with the latest developments in the deep learning research domain.

Scikit-Learn

Scikit-Learn is a user-friendly, production-ready python framework that implements numerous machine learning algorithms and tools and it is a remarkable starting point for both beginners and experts to master machine learning algorithms (Géron, 2019).

3.3 Data Preparation

The data preparation includes preprocessing and enhancement procedures to improve the quality and variety of the dataset.

3.3.1 Converting Image Data

In order to simplify compatibility and facilitate loading the existing mammography image data, a conversion from the DICOM data format to a Portable Network Graphics (PNG) data format was carried out for two reasons. Firstly, the additional information, like patient name etc., that may be contained in the DICOM format is not relevant to the scope of this thesis and it was already removed during the pseudonymization process of the image data. Therefore, the DICOM format proves to be unnecessary for the scope of this thesis. In addition, no data or fidelity was lost during the conversion process, maintaining the original brightness range of 0 to 255. Secondly, the PNG data format was chosen because it proved to be extremely compatible during the development process of the neural network. In contrast, the DICOM format is not widely supported by common code commands used in neural network development, which requires numerous workaround codes to be able to work with this format.

PNG is a widely used, standard lossless data format for images, primarily designed for graphics and supports functions such as transparency, different color modes, and metadata (Bill & Berger, n.d.).

The conversion of the data formats was carried out in a Jupyter Notebook web application from the Anaconda® distribution.

A function was created to convert the image data, which can be seen in listing 1 below.

```
def dicom_to_png(dicom_file_path, output_folder):
    # Read DICOM file
    dicom_data = pydicom.dcmread(dicom_file_path)
    # Extract pixel data
    pixel_data = dicom_data.pixel_array
    # Rescale pixel values to [0, 255] for uint8 type
    pixel_data_scaled = ((pixel_data - pixel_data.min()) /
      (pixel_data.max() - pixel_data.min()) * 255).astype(np.uint8)
    # Convert pixel data to PNG image
    image = Image.fromarray(pixel_data_scaled)
    # Save PNG image
    png_file_path = os.path.join(output_folder,
      f"{os.path.basename(dicom_file_path)[:-4]}.png")
    image.save(png_file_path, format="PNG")
```

Listing 1 Function `dicom_to_png`.

The listing 1 shows the codeline `pixel_data_scaled` which rescales pixel values in an array to fit within the range of [0, 255]. This is necessary when working with image data to ensure that pixel values fall within the valid range for the 8-bit unsigned integer (uint8) data type, which is commonly used to represent image pixels.

For the data type uint8 the values range of the pixels in the image from [0,255] (Kumar & Verma, 2010).

This function was called in a loop, which can be seen in listing 2, in which the code iterates over each file in the specified folders and executes the function for the files.

```
for dicom_filename in os.listdir(dicom_folder_path):
    dicom_filepath = os.path.join(dicom_folder_path,
      dicom_filename)
    # Skip non-DICOM files
    if not dicom_filename.lower().endswith('.dcm'):
        continue
    # Convert DICOM to PNG
    dicom_to_png(dicom_filepath, png_folder_path)
```

Listing 2 Loop through DICOM files and convert to PNG.

3.3.2 Understanding Data Representation

As Müller & Guido (2016) posit, visualising the data as a table can facilitate a more intuitive comprehension of the data employed in the machine learning process. On the one hand the rows would be the individual items so here the several mammograms and each of those entities in machine learning is referred to as a data point or sample. On the other hand, the columns would be the BI-RADS® classification which describe the properties of the images and therefore in machine learning be called features. For each BI-RADS® classification, from one to five, one column is representative. Multiplying the number of samples by the number of features will put out the shape of the data array.

As described by Badillo et al. (2020) the features in a dataset can be categorized as either categorical (predefined values without a specific order, such as female and male), ordinal (predefined values with a specific order, such as disease stages), or numeric (such as real quantitative values).

The possible outputs – which are the different BI-RADS® classifications – are called classes. Thus, every mammogram in the dataset belongs to one of the five BI-RADS® classes. The preferred output for a single data point – as mentioned before the data point is a single mammogram – is the corresponding BI-RADS® classification which the single mammogram belongs to, and this is called label. A label is a specific category assigned to an individual data point.

3.3.3 Normalize the Data

As outlined by Ghosh et al. (2020) data preprocessing is the process of transforming raw data into a more feature-rich, cleaner, and consistent format, thereby enhancing its learnability. This process occurs before feeding the data into the CNN model.

Geras et al. (2018) constructed a multi-view CNN that employs large high-resolution images without any form of compression. The study demonstrated that the precision of mammogram recognition and classification improves significantly with the size of the training dataset. Consequently, the optimal performance can only be reached when images are employed at their original resolution.

The mammography images used in this thesis have varying height and width formats due to post-processing steps after image capture. After the mammogram is taken, radiology technologists usually crop the images tailored to the relevant structures. This results in different image formats for the 1,480 extracted mammograms. To standardize the images, code was used in the Jupyter Notebook to ensure uniformity. The largest image format dimension was determined first, as

shown in listing 3. The provided code defines a function called `find_largest_dimension(folder_path)`, which takes a folder path as input. Within the function, it iterates over the folders and files in the specified directory. For each image file found, it opens the image using the PIL module, retrieves its width and height, and updates the `max_width` and `max_height` variables if the current image's dimensions exceed the previously recorded maximum dimensions. Finally, it returns the maximum width and height found in the entire folder structure. For the image data used in this thesis the largest image format dimension is: 3328 x 4096.

```
def find_largest_dimension(folder_path):
    max_width = 0
    max_height = 0
    for folder in os.listdir(folder_path):
        subfolder_path = os.path.join(folder_path, folder)
        for filename in os.listdir(subfolder_path):
            image_path = os.path.join(subfolder_path, filename)
            with Image.open(image_path) as img:
                width, height = img.size
                max_width = max(max_width, width)
                max_height = max(max_height, height)
    print("The largest dimension is:", max_width, "x", max_height)
    return max_width, max_height
```

Listing 3 Function to find the largest dimension.

Afterwards the mammograms were adjusted to the largest dimension by filling the missing part with black space, as seen in the listing 4. This method was chosen to ensure that no relevant structures were cut off and to avoid downsampling the images. The provided code defines a function called `adapt_to_largest_dimension_ratio` that adjusts images in a folder to the largest dimension found within that folder. The function first determines the largest dimension ratio by calling the `find_largest_dimension()` function, which returns the maximum width and height of images in the specified folder path. It then iterates through each subfolder and image within the input folder. The function determines the dimension ratio of each image and compares it with the largest dimension ratio found earlier. If the image's dimension ratio is less than or equal to the largest dimension ratio, the image is resized to fit that dimension ratio by creating a new blank image and pasting the original image onto it. Finally, the adapted image is saved to an output folder. If an image's dimensions exceed the largest dimension ratio, it is added to the `missing_images` list. The function

returns a list of filenames for any images that could not be adapted. This statement was included for security reasons to ensure that all images conform to the new dimensions.

```
def adapt_to_largest_dimension_ratio(folder_path, output_folder):
    max_width, max_height = find_largest_dimension(folder_path)
    largest_dimension_ratio = max_width / max_height
    black_color = 0
    missing_images = []
    for folder in os.listdir(folder_path):
        subfolder_path = os.path.join(folder_path, folder)
        output_subfolder_path = os.path.join(output_folder, folder)
        os.makedirs(output_subfolder_path, exist_ok=True)
        for filename in os.listdir(subfolder_path):
            image_path = os.path.join(subfolder_path, filename)
            with Image.open(image_path) as img:
                width, height = img.size
                dimension_ratio = width / height
                if dimension_ratio <= largest_dimension_ratio:
                    # Calculate new dimensions
                    new_width = int(height * largest_dimension_ratio)
                    # Create new blank image with the largest dimension
                      ratio
                    new_img = Image.new("L", (new_width, height),
                      black_color)
                    # Paste the original image onto the new blank image
                    paste_location = ((new_width - width) // 2, 0)
                    new_img.paste(img, paste_location)
                    # Save the adapted image
                    output_file_path =
                      os.path.join(output_subfolder_path, filename)
                    new_img.save(output_file_path)
                else:
                    missing_images.append(filename)
    return missing_images
```

Listing 4 Function to adapt images to the largest dimension ratio format.

Figure 10 illustrated the manner in which the image data changes before and after the adjustment to the largest image dimension. The image on the left is the original image, while the image on the right has been resized to the largest dimension. The

largest dimension, as seen on the right-hand side, was subsequently employed as the standard for all image data.

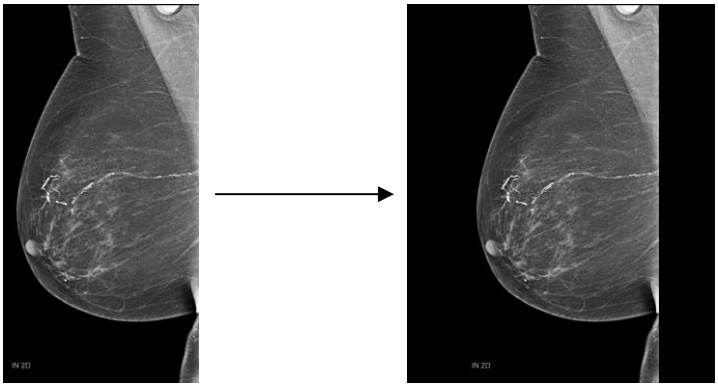

Figure 10 Comparison of the image data before and after adaption to the largest image dimension. Here is an example using mammography images from BI-RADS® two with the original dimension on the left and the standard dimension on the right side. Own illustration based on Ordensklinikum Linz GmbH Barmherzige Schwestern, 2023.

3.4 Model Architecture

The development of an effective CNN model for mammography image classifications represents a significant phase of the overall process. This includes the selection of an appropriate CNN architecture and the optimization of the model for the specific task. The objective is to achieve high levels of accuracy.

The deep learning model consists of many layers that capture the features of an image in depth (Mohi ud din et al., 2022). The model learns and extracts high-level features, with lower abstractions, in the first layers; and low-level features, with higher abstractions, in deeper layers (Ghosh et al., 2020).

As outlined by Mohi ud din et al. (2022) generic attributes, such as edges or circles, are extracted in the first layer. As illustrated in figure 11 the middle layer extracts middle-level features such as mouth, nose or eyes. The last layer extracts high-level features – these are objects such as the head.

Figure 11 Deep learning feature extraction method. Own illustration based on Mohi ud din et al., 2022, p. 6.

According to Ghosh et al. (2020) the organization of neurons in a cat's brain is also layered. In order to recognize visual samples, these layers learn by extracting local features and then combining them for a higher-level representation. This principle has become one of the main principles of deep learning.

A basic CNN structure, as described by Abdelhafiz et al. (2019), consists of a series of layers, including convolutional layers, non-linear layers, pooling layers and a loss function applied to the final fully connected layer. The outcome of the network can represent a single class or a probability distribution across classes that characterize an image best.

Ghosh et al. (2020) outline that a CNN usually consists of a series of blocks, each containing a convolutional layer and a pooling layer as visualized in figure 12. The blocks are followed by one or more fully connected layers and a final output layer. An image classification CNN comprises two main components: feature extraction and classification. The convolutional and pooling layers perform feature extraction, while the fully connected layer is responsible for classification.

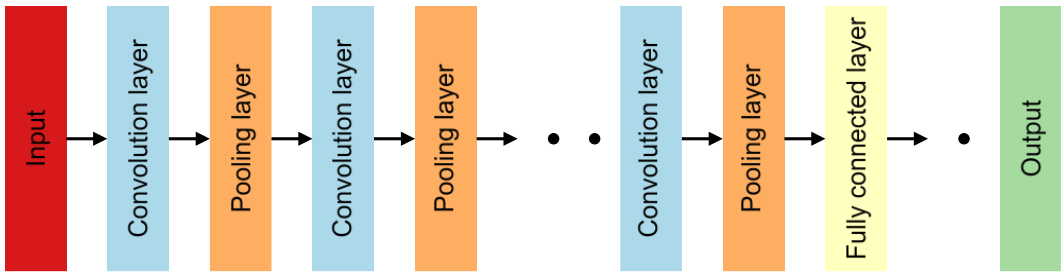

Figure 12 Theoretical model of CNN. Own illustration based on Ghosh et al., 2020, p.5.

According to Mohi ud din et al. (2022) the basic architecture of a CNN includes the following components:

- Multiple Convolution layers (filters/kernels),
- Activation functions,
- Pool layers (down-sampling),
- Fully connected layers (Dense layers).

In addition to the layers, there are other factors that influence the development of a CNN model, such as the activation function, normalization method, loss function, regularization, optimization, and processing speed (Ghosh et al., 2020).

3.4.1 Convolutional Layer

As mentioned by Mohi ud din et al., (2022) the convolutional layer uses filters. If a convolutional layer is used in a CNN, a filter – also known as a kernel – is placed over an input image and a mathematical convolution operation is performed.

Ghosh et al. (2020) state that the convolutional layer is the fundamental component of a CNN. Its purpose is to learn the feature representations of the input, which in this case are mammography images. Abdelhafiz et al. (2019) explain that the convolution layer comprises multiple learnable convolution kernels, also known as filters, that are convolved with the input image to generate an output feature map. Here, each convolution layer uses multiple filters to extract various types of characteristics. A kernel is a matrix of numbers where each value represents the weight of the kernel. In a CNN model, the weights of a kernel are randomly assigned at the beginning and then adjusted per training epoch. This allows the kernel to learn how to extract features.

Mohi ud din et al. (2022) posit that the convolutional layer applies a filter to the input image, with each region that the filter passes through, a single value is obtained by performing a mathematical convolutional operation. The process of the convolutional operation extracts features from an input image, with a particular emphasis on specific features in different parts of the image.

As described by Ghosh et al. (2020) the CNN input can be an image with multiple channels. For instance, an RGB image is composed of three channels, whereas a greyscale image, such as those used in this thesis for mammography, has only one channel. Figure 13 demonstrates the convolution operation with a greyscale image featuring a 4 x 4 grid and a 2 x 2 kernel with randomly initialized weights. During the convolution operation the 2 x 2 kernel is moved both horizontally and vertically throughout the entire 4 x 4 image. In the course of this process, the values are multiplied and added together to create a dot product between the kernel and input image. This generates a scaling value in the output feature map. The procedure is repeated until the kernel is no longer capable of being moved.

In figure 13 the convolution operation is performed without padding to the input image and with a stride to the kernel of one. The stride is the step width along the horizontal or vertical position. However, other stride values can be used. Increasing the stride leads to a lower-dimensional feature map. To prevent the features at the edges of the image from blurring too quickly, padding is used to give meaning to the information about the edge size. If padding is used to enlarge the size of the input image, the size of the output feature map is also increased.

By a receptive field each unit of a feature map is connected to the previous layer. A receptive field, also known as such in medicine, is the area that transmits information to a single downstream unit. To create a new feature map, the input is convolved with the kernels and then a non-linear activation function is applied to the convolved result for each element. The convolutional layer's property of sharing parameters means that the same set of weights is used at different spatial locations of the input data. The weight sharing function is one of the advantages and main reasons for considering a CNN. This reduces the number of individual trainable parameters that need to be learned compared to a fully connected layer, where each neuron has its own set of weights. By sharing parameters, the convolutional layer can capture local patterns and features in the input data while significantly reducing the overall complexity of the model. This leads to a more efficient use of parameters, computational resources and helps to avoid overfitting and improve the generalization of the model.

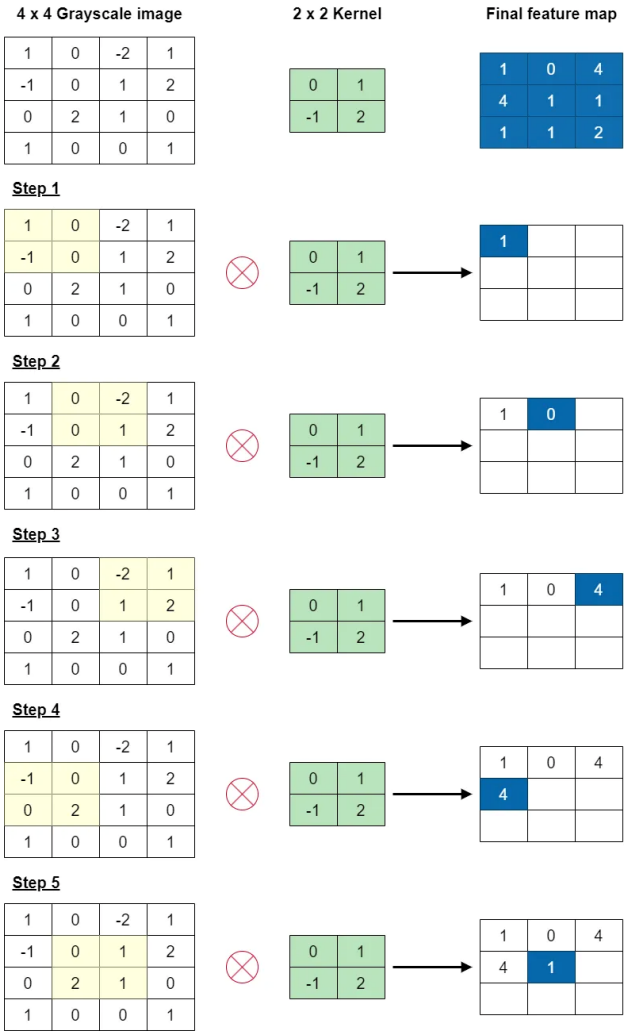

Figure 13 Visualization of the first five steps of convolution operation. Own illustration based on Ghosh et al., 2020, p.7 f.

3.4.2 Activation Function

As outlined by Ghosh et al. (2020), the primary function of activation functions in neural network models is to map inputs to outputs. Inputs are calculated by taking the weighted sum of the neuron's input and adding present bias. The activation function also determines whether the neuron should fire or not, which means if it should send a signal to the next layer or not. The CNN architecture employs nonlinear activation layers after each learnable layer, such as the convolutional and the fully connected layers. Furthermore, an activation function must be differentiable to measure how each part of the neural network contributes to errors. This implies that the activation function enables error backpropagation for the purpose of training the model. Activation functions used in neural network architectures include mathematical functions such as sigmoid, rectified linear unit (ReLU), tanh, and softmax functions.

According to Géron (2019) it is important to note that the softmax regression classifier can only predict one class at a time. It is typically used in situations where there are multiple different categories, but each instance belongs to only one category, as is the case in the model described in this thesis. In this case, a mammography image can also be assigned to a single BI-RADS® classification.

Müller & Guido (2016) outline that typically the ReLU activation function, also called rectifying nonlinearity, is used. The specialization of the ReLU function is, that it cuts off the values below zero, visualized as a mathematical graph in figure 14. This means that it converts all input values into positive numbers and compared to other methods the ReLU function has the advantage of necessitating minimal computation load (Ghosh et al., 2020).

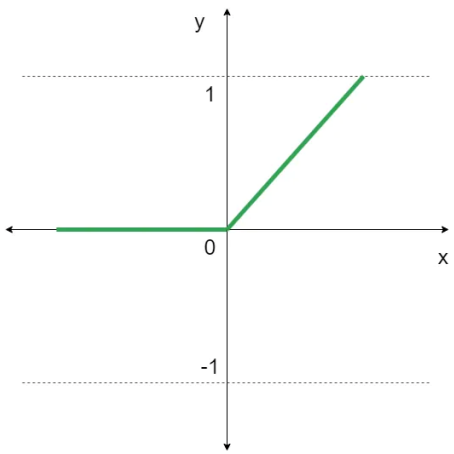

Figure 14 ReLU activation function. Own illustration based on Ghosh et al., 2020, p.11.

The ReLU function is widely used in practice and has become the default due to its computational efficiency as outlined by Géron (2019). However, one disadvantage of the ReLU activation function is the problem of dying ReLUs. During training, some neurons may output only zero and become inactive. In cases with a very high learning rate, it is possible that half of the neurons in the network become inactive. To address this issue, the leaky ReLU function, a variant of the ReLU function, can be employed. This process ensures that neurons remain active to some extent, rather than becoming completely inactive.

Moreover the activation function, especially the ReLU, is applied after each convolution layer to avoid the vanishing gradient problem (Mohi ud din et al., 2022).

The vanishing gradient problem occurs due to the backpropagation algorithm as mentioned by Géron (2019). The backpropagation algorithm determines the contribution of each component of the neural network to the final result. It starts at the output and adjusts backward to the input, to reduce errors. Unfortunately, as the algorithm progresses deeper into the lower layers, the gradients often become increasingly smaller. This makes it challenging for the network to learn effectively. It is comparable to attempting to correct an error by implementing minuscule adjustments that fail to result in any significant alteration.

Ghosh et al. (2020) describes the vanishing gradient problem as follows: during backpropagation of a CNN with many layers, for example 1,000, the gradient of loss, also called error, must be calculated with respect to the corresponding weights in the neurons of each layer to update these weights. This process uses the derivative operation. As a result, the gradient become smaller and smaller the further backward the network is moved. Consequently, the neurons in the earlier layers receive tiny gradients, some of which approach zero. This results in the preceding layer’s gradient being updated to a minimal extent, which in turn leads to a slow and inefficient learning process.

In contrast, the exploding gradient problem arises when the adjustments become so significant as the algorithm progresses in reverse that the network's weights become excessively large, resulting in the entire system becoming uncontrollable (Géron, 2019).

3.4.3 Pooling Layer

The pooling layer is used for dimensionality reduction and is based on the "sliding window" principle (Mohi ud din et al., 2022). In the pooling or also called sub-sampling layer, a small region of the convolution output is used as input and reduced in size to produce a single output (Ghosh et al., 2020). This layer reduces

the number of required parameters and operations – hereby achieving a down-sampling of the feature map – and helps to reduce the computational complexity of the network (Mohi ud din et al., 2022).

Ghosh et al. (2020) mentioned that like the convolution operation, the pooling operation is carried out by specifying the size of the pooled area and the stride of the operation. The pooling layer assists the CNN to determine whether a particular feature is present in the input image or not. However, the pooling layer does not consider the precise location of this feature, which can sometimes reduce the performance of the CNN. Various pooling techniques exist, such as max-pooling, min-pooling, sum-pooling, and average-pooling.

As outlined by Mohi ud din et al. (2022) the max-pooling layer is the preferred choice. For this pooling layer, the maximum value within a pooling window is selected and these conspicuous features are highlighted.

3.4.4 Fully Connected Layer

The final part of a CNN for classification typically comprises one or more fully connected layers as mentioned by Ghosh et al. (2020). This layer is a type of feed-forward ANN that follows the tenets of the traditional MLP neural network. The final convolution or pooling layer provides input to the fully connected layers in the form of a set of metrics known as feature maps. The feature map is flattened to be represented as a vector and placed in the fully connected layer. Ultimately the fully connected layer is the output layer – the classifier – of the CNN. This means that the final output of the model is generated by the fully connected layer.

Per explanation of Mohi ud din et al. (2022) the fully connected layer classifies the features extracted from the convolutional layer and the pooling layer. Here, all nodes of the layer are connected to the nodes of the previous layer, as visualized in the architecture of the fully connected layer shown in figure 15. In the last dense layer, there is a single artificial neuron for each target class. And for each of these neurons a probability value between zero and one is generated by using the softmax activation function.

As can be seen in figure 15 and described by Géron (2019), one output neuron is required per class, provided that each instance can only belong to a single class. The softmax activation function should be used for the entire output layer. Due to the softmax function, all estimated probabilities lie between zero and one and add up to one. This is necessary in a so-called multiclass classification where the classes exclude each other.

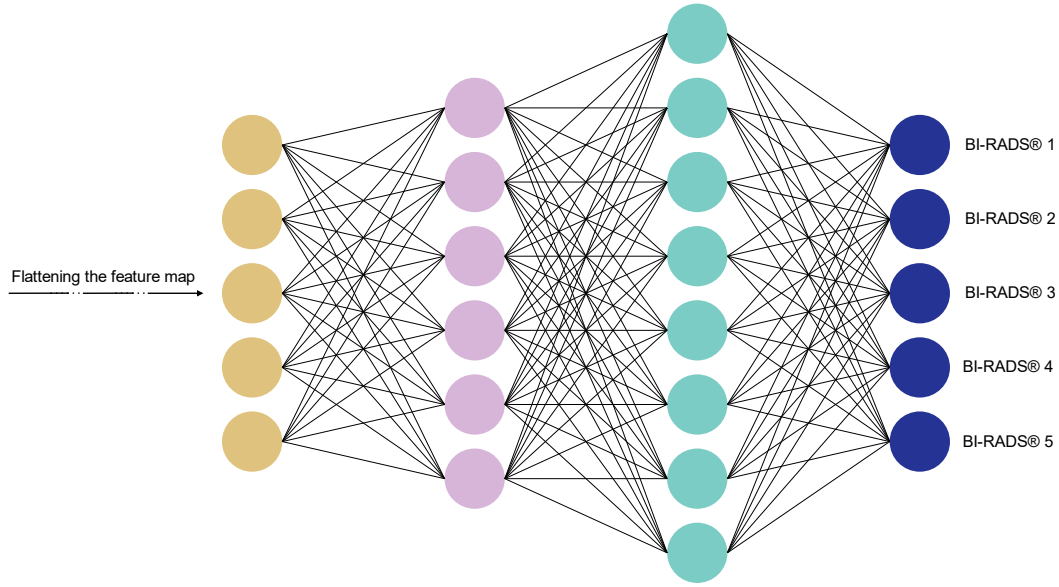

Figure 15 Architecture of the fully connected layers. Own illustration based on Ghosh et al., 2020, p.13.

3.4.5 Loss Function

Badillo et al. (2020) outline that the objective of model fitting is to obtain optimal parameter values, such as coefficients and weights, by defining and minimizing the distance between the model and the data. This distance is also referred to as the loss function or cost function. On the one hand, the model fitting involves ensuring that the projected outcomes of the model closely align with the actual data points in the training dataset. And on the other hand, the model should have the capability to generalize well and extend its predictions beyond the training dataset.

According to Ghosh et al. (2020) the output layer represents the final layer in the architecture of a CNN, where the final classification occurs. In the output layer, the loss function calculates the prediction error generated across the training samples. This error indicates the deviation between the predicted and actual outputs, and it is refined during the learning process. To calculate the error, two parameters are used: the estimated output of the model, also known as the prediction, and the actual output, which is also known as the label. Different types of loss functions are used for different types of problems, such as the cross-entropy, the euclidean loss function, or the hinge loss function.

In multi-class classification problems Ghosh et al. (2020) outline that the performance of the CNN model is often measured using the cross-entropy loss, also called softmax loss or log-loss function. The output is represented by the

probability p. The softmax activation function is used in the output layer to generate the output within the probability distribution.

Géron (2019) suggests that the cross-entropy loss is commonly advisable.

The euclidean loss is used for regression problems, whereas the hinge loss function is used in binary classification models (Ghosh et al., 2020).

3.4.6 Optimizer

According to Ghosh et al. (2020) the learning process of the model involves two main aspects: the selection of the learning algorithm, the so-called optimizer and applying various improvements to optimize the results. The principal goal is to minimize the error indicated by the loss function, which is the difference between the predicted and the actual outputs. The loss function is based on adjustable parameters such as weights, biases, etc. Naturally, gradient-based learning methods are used for CNN models. During each training iteration, the model parameters are constantly improved to decrease the error and search for the ideal outcome. The learning rate determines the number of steps required to update the parameters. A complete iteration of the parameter update over the entire training dataset is referred to as a training iteration. Among the various types of gradient-based learning algorithms, the most commonly applied ones are the batch gradient descent, the mini batch gradient descent, and the stochastic gradient descent.

Gradient descent is an algorithm that iteratively updates an objective function by adjusting parameter values in the direction of the steepest descent of the objective function until an approximation to a minimum distance is achieved (Badillo et al., 2020). As stated by Géron (2019), the initial step is the random initialization, which populates the parameters with random values. The algorithm is then gradually improved step by step by minimizing the cost function until it reaches to a minimum. The size of the steps in the gradient descendent is regulated by the learning rate. For a too low learning rate, the algorithm requires many iterations to approach a minimum, resulting in increased time consumption. Conversely, with a very high learning rate, the algorithm may jump over the minimum valley and diverge with increasingly larger values, failing to find a good solution. This may result in an unfavorable divergence of the loss function (Abdelhafiz et al., 2019).

The model described in this thesis employs the Adaptive Moment Estimation (Adam) learning strategy, which adaptively calculates the learning rate for each parameter in the network as explained by Ghosh et al. (2020). Adam combines the advantages of two other learning strategies, Momentum and Root Mean Square Propagation (RMSprop).

3.4.7 Model Definition and Compilation

The code presented in listing 5 outlines the sequential structure of layers that include convolutional and pooling operations, complemented by dropout regularization, and results in a fully linked layer for classification into two classes: benign and malicious. It is important to note that for the classification into three or five classes, rather than two, the `num_classes` variable at the beginning of the code must be modified to `num_classes = 3` or `num_classes = 5`. Furthermore the softmax activation function is used in the final dense layer `layers.Dense(num_classes, activation='sigmoid')` and the loss has to be modified to `loss = tf.keras.losses.SparseCategoricalCrossentropy (from_logits=False)`. Finally, the model is compiled with appropriate loss and optimization functions to facilitate training and evaluation.

```
num_classes = 1
# Define the model layers
model = Sequential([
    layers.Input(shape=(img_height, img_width, 1)),
    # Convolutional layers
    layers.Conv2D(16, 3, padding='same', activation='relu',
      input_shape=(img_height, img_width, 1)),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    # Dropout layer
    layers.Dropout(0.2),
    # Flatten layer and dense layer
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes, activation='sigmoid')
])
# Compile the model
model.compile(optimizer='adam',
  loss=tf.keras.losses.BinaryCrossentropy(from_logits=False),
  metrics=['accuracy'])
model.summary()
```

Listing 5 Model definition and compilation with summary.

3.5 Training

Models are created using the training dataset as mentioned by Badillo et al. (2020). The validation dataset is then used to select the algorithm and, if necessary, the hyperparameters. The model with the best performance in the validation dataset is selected.

One of the most significant challenges in training CNNs is the size of the training dataset as mentioned by (Abdelhafiz et al., 2019). There are various strategies for addressing this issue, including data augmentation, transfer learning, and dropout. Nevertheless, a limited sample size still represents a considerable obstacle for training.

3.5.1 Validation

The training data is utilized to train multiple models with varying parameters as explained by Géron (2019). However, if a part of the training data is excluded beforehand, a validation dataset can be created to assess multiple models and select the optimal one. The model that performs the best on the validation set is then chosen. Finally, this selected model is evaluated on the test set to estimate the generalization error.

As outlined by Müller & Guido (2016) an independent dataset is required to evaluate the model, but it should not be used to build the model, unlike the training dataset. One approach to achieving this is by dividing the data into three distinct sets, as illustrated in figure 16: the training set for model creation, the validation set (or development set) for parameter selection, and the test set for evaluating the performance of the selected parameters.

Figure 16 Data divided into three parts: a training set, a validation set, and a test set. Own illustration based on Müller & Guido, 2016, p. 262.

In the CNN described in this thesis, the validation dataset was split off from the training data, shown in listing 6. Accordingly, 20% of the training dataset is used for validation.

```
img_height = 274
img_width = 224
batch_size = 32
```

```
train_ds = tf.keras.preprocessing.image_dataset_from_directory (
  data_dir,
  validation_split = 0.2,
  subset = "training",
  seed = 42,
  image_size = (img_height, img_width),
  batch_size = batch_size,
  color_mode="grayscale"  # Specify grayscale color mode
)
val_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  validation_split = 0.2,
  subset = "validation",
  seed = 42,
  image_size = (img_height, img_width),
  batch_size = batch_size,
  color_mode="grayscale"  # Specify grayscale color mode
)
```

Listing 6 Generating the training and validation dataset.

3.5.2 Data Augmentation

According to Abdelhafiz et al. (2019), data augmentation is a technique used to artificially enlarge the size of the training dataset. It is important to emphasize that data augmentation should only be applied to the training data. It involves applying various operations to the existing data samples, which results in one or more new versions of the data samples. Afterwards these new versions are used in the training process. In many real-life scenarios, particularly for medical datasets, there is often a limited amount of training data available. Therefore, data augmentation plays a crucial role in improving the CNN model. Various methods for data augmentation exist, such as cropping, rotations, flipping, translation, contrast adjustment, scaling and much more. These operations can be performed individually or in combination to increase the amount of training data. The advantages of data augmentation include the reduction of overfitting, increased generalization of the model, and enhanced performance.

The models in this thesis are trained with and without applying data augmentation. For the data augmentation the code in listing 7 is used. There are three data augmentation layers added sequentially to the model. The first layer flips the input horizontally. The second layer applies a random rotation of the input image with a

maximum rotation angle of 0.1 in radians. And the third layer randomly zooms in or out on the image with a specified maximum zoom of 10%.

Furthermore, the code of the model definition in listing 5, is extended with `data_augmentation` to apply the data augmentation.

```
data_augmentation = tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal",
      input_shape=(img_height, img_width, 1)),
    layers.experimental.preprocessing.RandomRotation(0.1),
    layers.experimental.preprocessing.RandomZoom(0.1)
])
```

Listing 7 Definition of the data augmentation pipeline.

3.5.3 Class Weights

According to (Abdelhafiz et al., 2019), another issue in the training process is the imbalanced ratio between the classes in the training dataset. The training of CNN models on imbalanced datasets can result in a bias in the prediction towards the more frequent class. The existing literature contains examples of work with both balanced and imbalanced datasets.

When the distribution of classes in the training set is imbalanced, with some classes having more samples than others, according to Géron (2019) it is advisable to adjust the `class_weight` parameter while fitting the model, as shown in listing 8. This parameter assigns greater weight to underrepresented classes and lesser weight to overrepresented ones. Keras uses these weights during loss calculations.

```
from sklearn.utils import class_weight
train_labels = np.concatenate([y for _, y in train_ds], axis=0)
class_weights = class_weight.compute_class_weight(
    class_weight='balanced',
    classes=np.unique(train_labels),
    y=train_labels
)
```

Listing 8 Class weights for imbalanced data.

3.5.4 Overfitting and Underfitting

As asserted by Müller & Guido (2016) the objective is to construct a model that can generalize with the highest possible accuracy. And if a model can accurately

predict unseen data, it can generalize from the training set to the test set. To measure how well the algorithm performs on new unseen data the model is evaluated on the test set.

The ability to correctly adapt to new or previously unknown inputs is called generalization as explained by Ghosh et al. (2020). If the model generalizes well for both the training data and the test data, it is called a just-fitted model. The main problem that arises when trying to ensure that a CNN model achieves good generalization is overfitting.

According to Müller & Guido (2016) overfitting is the term used to describe a model that is too complex for the amount of information available. This phenomenon occurs when a model focuses and fits too closely to the training set. Resulting in a good performance on the training set but a poor or even hardly generalization to new data, so the model is not able to generalize to new data very well. Conversely if it is a too simple model, it may not be able to adequately capture all the variability and aspects in the data. The diversities in the training data cannot be captured by the model. Resulting in a poor performance even on the training dataset. This occurrence is referred to as underfitting. The challenge is to identify the optimal balance between the opposing forces of overfitting and underfitting. The more complex a model can be, the better the predictions for the training data will be. However, if the model focuses too much on each individual data point in the training dataset, the model becomes too complex, and it cannot be generalized well to new data. In addition, it is important to consider that the variant range of inputs in the training dataset is closely related to the model complexity. A larger dataset with a greater variety of data points allows the use of more complex models without overfitting. Therefore, gathering more datapoints typically provides more variety and allows the construction of a more complex model.

Figure 17 visualizes the three different types of models: over-fitted, under-fitted, and just-fitted.

The most effective method to prevent over-fitting is to train the model on a large and diverse dataset, which can be accomplished by data augmentation as stated by Ghosh et al. (2020). Regularization is a technique used to prevent overfitting by implementing several intuitive ideas. There are many different approaches for regularization, such as dropout, drop-weights, the i^1 regularization, the i^2 regularization, which is also called weight decay, and many others. One of the most frequently used methods of regularization is dropout.

Abdelhafiz et al. (2019) mentioned different key strategies to handle overfitting, like data augmentation, batch normalization, or the regularization method dropout. The application of these techniques has been demonstrated to reduce overfitting and enhance the generalization ability of CNN models.

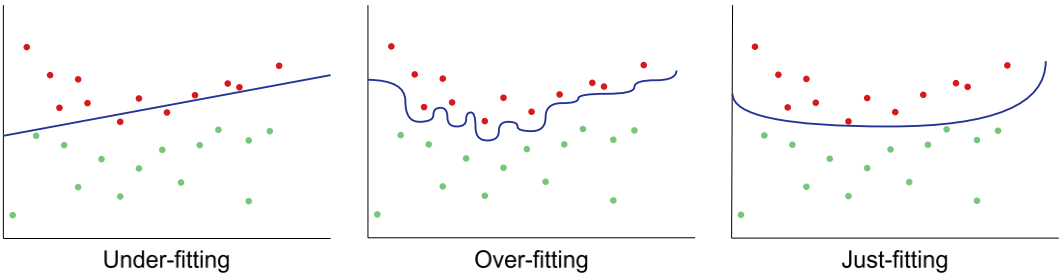

Figure 17 Examples of over-fitting, under-fitting and just-fitting by binary classification. Own illustration based on Ghosh et al., 2022, p. 19.

3.5.5 Dropout

Ghosh et al. (2020) describe the dropout regularization method, whereby during each training epoch neurons are randomly removed from the network. This removal of individual neurons is shown in figure 18. As some neurons are dropped, the goal is to equally distribute the power of feature selection across all units. This forces the model to learn multiple independent features. If a neuron is removed, this neuron does not participate in either forward or backward propagation during the training process. However, during the test process, the entire network with all neurons is used for prediction.

According to Géron (2019) in practical applications, dropout is typically implemented only to the neurons within the top one to three layers, with the exception of the output layer.

Smirnov et al. (2014) conducted a comparative analysis of regularization methods with CNNs, demonstrating that dropout is typically more effective than competing regularization techniques.

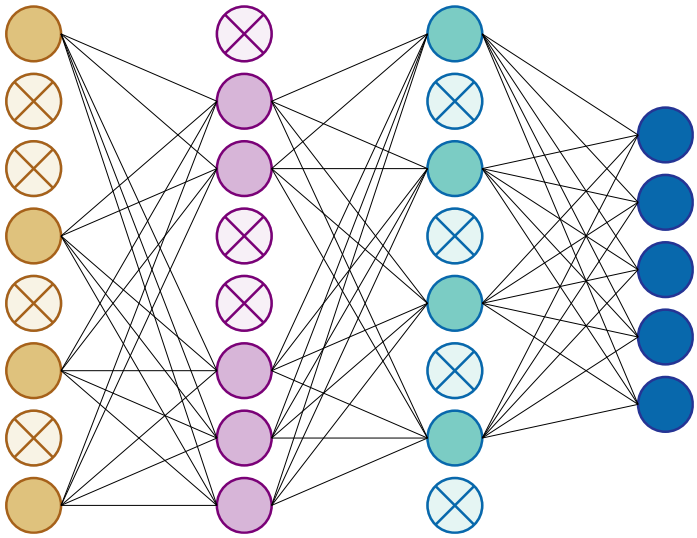

Figure 18 A neural network after applying dropout. Own illustration based on Ghosh et al., 2022, p. 19.

3.5.6 Model Training

The code in listing 9 outlines the training and evaluation of the developed CNN model. The code block calculates the number of steps per epoch based on the size of the training dataset and the batch size. An epoch is a complete run through the entire training dataset (Géron, 2019). This means that the code determines the number of iterations required to complete an epoch, which is a single pass through the entire training dataset, based on the size of the training dataset and the batch size used for training. Subsequently, the training phase of the model is carried out over a given number of epochs.

```
steps_per_epoch = len(train_ds) // batch_size
epochs = 20
history = model.fit(
    final_train_ds,
    validation_data=final_val_ds,
    epochs=epochs,
    class_weight=dict(enumerate(class_weights))
)
```

Listing 9 Model training.

Furthermore, as shown in listing 10 the training history, including accuracy and loss metrics, is presented in a visual format to enable the evaluation of the model performance during the training and validation phases.

```
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label='val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

Listing 10 Visualization of the training history.

3.5.7 Baseline Accuracy

In the field of machine learning the baseline accuracy represents the degree of accuracy that can be achieved even by a simple model as explained by Vaidya et al. (2024). It assists as a reference point for the evaluation of the performance of more complex models.

There are two possible baseline models as described by Lee (2021). The zero rate classifier (ZeroR) and the random rate classifier. The ZeroR model, which is used in this thesis, consistently predicts the most prevalent class within the dataset.

As the baseline accuracy can be used as a reference point to assess whether the developed model has fulfilled its purpose, it was also referred to in this thesis.

The formula for the baseline accuracy looks as follows:

$$Baseline\ Accuracy = \frac{m}{n}$$

- n instances in the dataset
- m instances in the majority class

For example, taken the binary classification presented in this thesis between benign and malignant: To the benign class belong 784 out of 1,182 images in the training dataset which represent 66.33%, whereas 398 images which represent 33.67% belong to the malignant class. A ZeroR model would always predict the benign class. In this case, the baseline accuracy is 66.33%, as the model would be correct 66.33% of the time.

In the case of multiclass classification with three classes, the normal class contains 518 images, representing 43.82% of the total. The benign class contains 226 images, representing 19.12% of the total, while the malignant class contains 398 images, representing 33.67% of the total. Consequently, the baseline accuracy is 43.82%.

For the multiclass classification task involving five classes: the BI-RADS® one class encompasses 266 images, representing 22.50% of the training dataset, the BI-RADS® two class includes 292 images, representing 24.70%, the BI-RADS® three class incorporates 226 images, representing 19.12%, the BI-RADS® four class compromises 194 images, representing 16.41% while the BI-RADS® five class encompasses 204 images, representing 17.26%. Consequently, the baseline accuracy is 24.70%.

In order for a model to demonstrate that it possesses the required capabilities for a given problem, it is essential to reach an accuracy higher than the baseline accuracy according to (Lee, 2021). Nevertheless, if the accuracy of the model is not significantly higher than the baseline accuracy, it is questionable whether the model offers any evident added value.

3.6 Testing

The test dataset is used to evaluate the generalisation error, which indicates the prediction error of the test dataset (Badillo et al., 2020).

According to Müller & Guido (2016) it is crucial to have a separate test dataset that is exclusively used for the final evaluation, when developing and evaluating a model. The test set is used to make predictions and evaluate the accuracy of the model. For all exploratory analyses and model selection it is recommended to use a combination of a training set and a validation set. Therefore, the test set should only be used for the final evaluation. If more than one model was evaluated on the test set, selecting the better model results in an overly optimistic assessment of the model's accuracy. For multiclass classification, the accuracy is defined as the fraction of correctly classified examples. If the classes are imbalanced the accuracy may not be a reliable evaluation measure. Therefore, this model includes a code for weighting the classes, as described in chapter 3.5.3.

The test set contains data that was not used for building the model, but the correct species for each sample in the test set is known. In the listing 11 the test dataset is generated similar to the training’s dataset.

```
test_ds = tf.keras.preprocessing.image_dataset_from_directory(
    test_data_dir,
    image_size=(img_height, img_width),
    batch_size=batch_size,
    color_mode="grayscale"  # Specify grayscale color mode
)
```

Listing 11 Generate the test dataset.

The accuracy of the model is measured by comparing the predicted classification for each mammogram in the test data against its actual label. The accuracy is calculated as the fraction of mammograms for which the correct label was predicted. So, if the classification performance of the model is evaluated using the accuracy, it gives out the fraction of correctly classified mammograms. The code in listing 12 evaluates the model's performance on the test dataset and print the obtained test loss and accuracy.

```
test_loss, test_accuracy = model.evaluate(final_test_ds)
print("Test Loss:", test_loss)
print("Test Accuracy:", test_accuracy)
```

Listing 12 Evaluation of the model performance on the test dataset.

It is common for the test set to show slightly lower performance compared to the validation set, due to the fact that the hyperparameters are optimized based on the validation set, not the test set (Géron, 2019).

Due to the fact that the breast tissue does not completely overlap in the CC and MLO projections, the projections should not be labelled the same by default (Isosalo et al., 2023).

In order to gain a more comprehensive understanding of the evaluation of the respective models, additional evaluation methods have been employed.

3.6.1 Receiver Operating Characteristic and Area Under the Curve

According to Songsaeng et al. (2021), the ROC curve is a graphical representation of the sensitivity and specificity of a diagnostic model at different thresholds. The sensitivity is the true positive rate (TPR), whereas the specificity is the true negative rate (TNR). The ROC curve is used to describe the diagnostic capability of the model.

The area under the ROC curve (AUROC, AUC) is a single metric that indicates the overall discriminatory power of the model as outlined by Narkhede (2018). It is a measure of the extent to which the model is able to differentiate between the classes. An AUROC of 100% corresponds to a classifier whose predictions are perfectly accurate.

The ROC curve and the AUROC are effective diagnostic tools for binary classification problems in machine learning. The code in listing 13 provides insight into the evaluation of the model according to the ROC, which is used for the binary classification model.

```
true_labels = []
for images, labels in final_test_ds:
    true_labels.extend(labels.numpy())
# Predict probabilities for the test dataset
y_pred_prob = model.predict(final_test_ds)
# Compute false positive rate (fpr), tpr, thresholds and roc_auc
fpr, tpr, thresholds = roc_curve(true_labels, y_pred_prob)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(8, 8))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=
  'ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC)')
plt.legend(loc="lower right")
plt.text(0.6, 0.2, 'AUC = %0.2f' % roc_auc, bbox=dict(facecolor='white',
  alpha=0.5))
plt.show()
```

Listing 13 Evaluation of the binary classification model using ROC and AUC.

3.6.2 Confusion Matrix

Kundu (2022) explains that a confusion matrix is a table which provides an overview of the classification performance of the model using the test set with known true values. The classification model's predictive performance is presented here as a class-wise distribution. This enables the identification of areas where the model is uncertain or incorrect.

The confusion matrix is a tool for comparing the actual target values against the model's predictions as outlined by Suresh (2020). In this context, the instances of the actual class are represented by the rows of the matrix. Conversely, the instances of the predicted class are represented in the columns, or vice versa. The confusion matrix is designed to differentiate between various terminological categories.

- True Positive (TP): Case where the model has made a positive prediction, and the truth is also positive.
- True Negative (TN): Cases where the model has made a negative prediction, and the truth is also negative.
- False Positive (FP): Cases where the model has made a positive prediction, but the truth is negative. This is also known as a type 1 error.
- False Negative (FN): Cases where the model has made a negative prediction, but the truth is positive. This is also known as a type 2 error.

The confusion matrix was employed in this thesis, as seen in the code in listing 14, to assess the multiclass classification models. In the context of a multiclass confusion matrix, the concept of positive and negative classes, as defined in the binary classification, is replaced by the individual classes of the problem. The predicted class, in the column, is counted in relation to its true class in the rows.

According to Binny & Omair (2024) the matrix offers a comprehensive analysis of a model's performance across multiple classes, thereby providing insights into its strengths and weaknesses. It serves as a valuable instrument for the assessment of classification models, particularly in scenarios with multiple classes.

```
predicted_probabilities = model.predict(final_test_ds)
predicted_labels = np.argmax(predicted_probabilities, axis=1)
true_labels_list = [label.numpy() for _, label in final_test_ds]
true_labels_flat = [item for sublist in true_labels_list for item in
  sublist]
conf_matrix = confusion_matrix(true_labels_flat, predicted_labels)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
  xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

Listing 14 Evaluation of the multiclass classification model using the confusion matrix.

4 Machine Learning Workflow

The flowchart in figure 19 illustrates the process flow involved in developing and evaluating a CNN. It includes important steps such as data preparation, model construction, training, and evaluation, providing a structured overview of the CNN workflow from input data acquisition to final performance assessment on the test dataset.

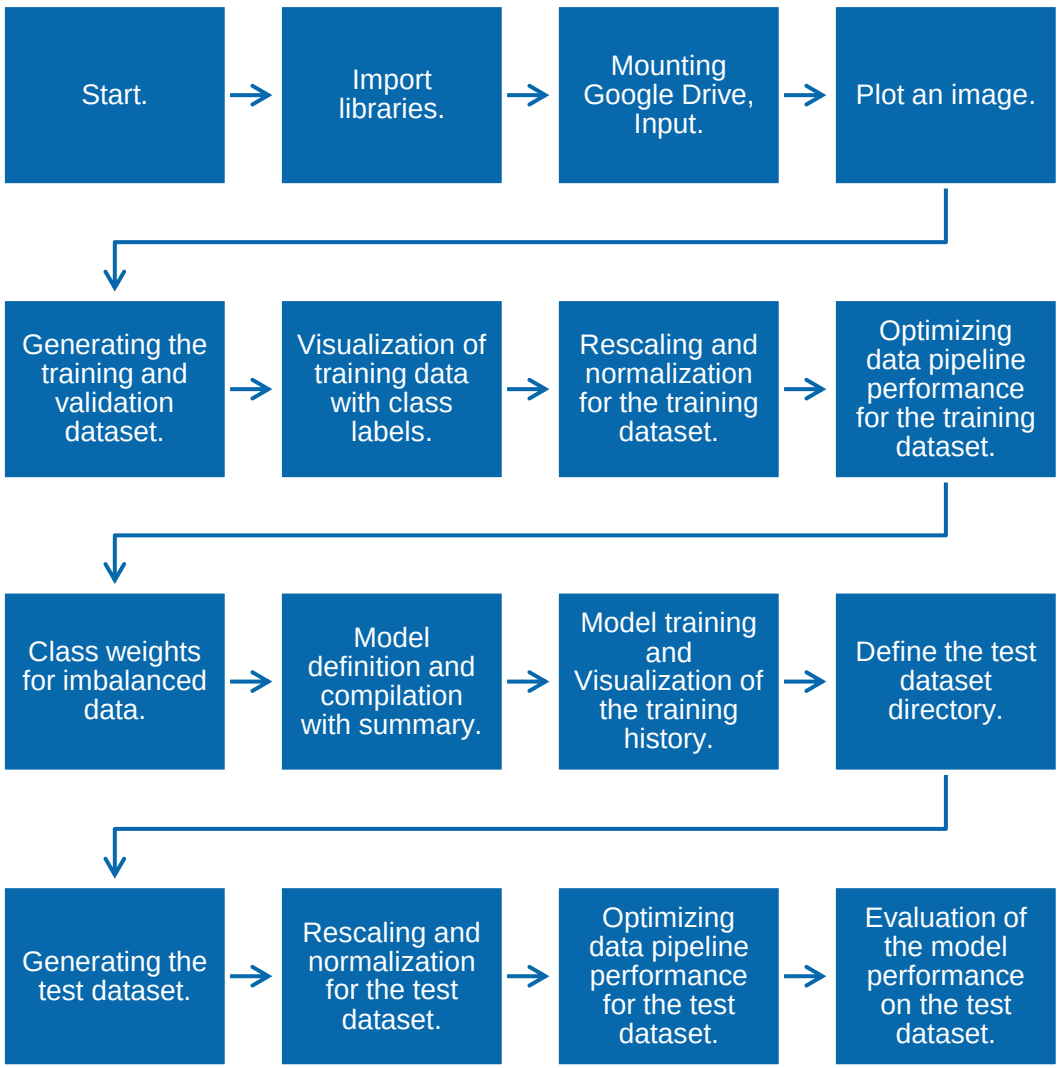

Figure 19 CNN workflow diagram. Own illustration.

After importing the libraries and modules used, as described in chapter 3.2.2, the mammography dataset, previously uploaded to Google Drive, is connected to Google Colab, as shown in the code in listing 15. In the last lines of code, the total number of images in the training dataset is calculated by counting the number of files with the extension ".png" in each subdirectory of the directory `data_dir` previously referenced by the training dataset.

```
from google.colab import drive
drive.mount('/content/drive');

data_dir = pathlib.Path
  ("/content/drive/MyDrive/Daten/Two_Mammograms_PNG_Split_train
  test_normalized/train")

image_count = len(list(data_dir.glob('*/*.png')))
print("Image Count for the training - and validation - dataset:",
  image_count)
```

Listing 15 Mounting Google Drive, input and counting images.

Following that, one image of each class – either the BI-RADS® classes, benign, malignant, or normal – is displayed for verification purposes, using the code shown in listing 16.

```
benign = list(data_dir.glob('benign/*'))
print (len(benign))
PIL.Image.open(str(benign[0]))
```

Listing 16 Plot an image.

Afterwards, the mammogram images of the previously referenced training dataset are divided into two groups, the training dataset and the validation dataset, as described in chapter 3.5.1.

In the following code block listing 17, the data is visualized again for visual inspection. First, the class names are extracted from the `train_ds` dataset in the code. The aspect ratio of the images is then calculated using the width and height dimensions. A diagram is then initialized with a size corresponding to the aspect ratio of the images. The code iterates over the first set of images and corresponding labels from the training dataset. In a loop, the first nine images of the batch are displayed as subplots in a three-by-three grid. Each subplot displays one image from the stack along with the corresponding class label.

```
class_names = train_ds.class_names
print(class_names)

aspect_ratio = img_width / img_height
plt.figure(figsize=(10, 10 * aspect_ratio))

for images, labels in train_ds.take(1):
  for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(images[i].numpy().astype("uint8"), cmap="gray")
    plt.title(class_names[labels[i]])
    plt.axis("off")
```

Listing 17 Visualization of training data with class labels.

The data is exposed to rescaling and normalization in order to ensure the consistency of the feature scales and to facilitate convergence when training the model. Rescaling brings the pixel values into a desired range, while normalization ensures a mean of zero and unit variance. The code provided in listing 18 is essential for standardizing the input data before feeding them into the CNN, as this can significantly improve training stability and convergence. The Rescaling layer scales pixel values of input images to the range between zero and one by dividing each pixel value by 255. This ensures that all pixel values fall within the same range, facilitating the learning process of the CNN and enabling it to converge efficiently. With the mapping, the normalization is applied to each image in the dataset while maintaining the corresponding labels unchanged. Similarly, this line applies the same normalization to the validation dataset. Finally, the last line prints out the minimum and maximum pixel values of the first normalized image. By ensuring that the pixel values are within the range zero and one, it can be confirmed that the normalization process has been successfully applied.

```
normalization_layer =
layers.experimental.preprocessing.Rescaling(1./255)
normalized_train_ds = train_ds.map(lambda x, y:
  (normalization_layer(x), y))
normalized_val_ds = val_ds.map(lambda x, y: (normalization_layer(x), y))
image_batch, labels_batch = next(iter(normalized_train_ds))
first_image = image_batch[0]
print(np.min(first_image), np.max(first_image))
```

Listing 18 Rescaling and normalization.

The data is prepared in advance to facilitate the training of the CNN. This is achieved by utilizing the code block in listing 19, which involves preparing the data in advance to speed up the training of the CNN. This ensures that the data is available at the appropriate time during the CNN process, which can speed up the training process. Autotune sets a value to decide how much data to load at one time, optimizing the efficiency. The value is automatically adjusted in accordance with the computer's processing capabilities. Subsequently, the training data `normalized_train_ds` is prepared, although not all of it is loaded simultaneously. Initially, only a portion of the data is loaded into memory while the CNN is still working on the previous part. This approach ensures that the training process runs smoothly and is accelerated. The same methodology is applied to the validation data `normalized_val_ds`. This ensures that the data is available when the validation dataset is required for the purpose of evaluating the efficacy of the CNN, without impeding the overall process.

```
AUTOTUNE = tf.data.experimental.AUTOTUNE
final_train_ds = normalized_train_ds.prefetch(buffer_size=AUTOTUNE)
final_val_ds = normalized_val_ds.prefetch(buffer_size=AUTOTUNE)
```

Listing 19 Optimizing data pipeline performance.

Afterwards the classes were weighted as already described in chapter 3.5.3, following by the model definition, as outlined in chapter 3.4.7 and the training process of the CNN, mentioned in chapter 3.5.6.

In order to assess the model using the test dataset, the test dataset, in a manner similar to the training dataset, is first imported from Google Drive into the Google Colab file, as illustrated in listing 20.

```
test_data_dir =
  /content/drive/MyDrive/Daten/Two_Mammograms_PNG_Split_train
  test_normalized/test'
```

Listing 20 Define the test dataset directory.

The test dataset is then generated in a manner comparable to the training dataset. Similarly, the rescaling, normalization and the optimization of the data pipeline performance is performed. Finally, the model is evaluated using the test dataset, as indicated in chapter 3.6.

4.1 Binary versus Multiclass Classification Model

A comparison of the code for implementing the binary classification model and the multiclass classification model reveals some differences.

The binary classification model employs the binary cross-entropy loss function, which is well-suited to binary classification tasks where the output is either zero or one. In contrast, the multiclass classification model employs the sparse categorical cross-entropy loss function, which is optimal for multi-class classification tasks where each sample belongs to one of multiple classes.

Regarding the activation function employed in the output layer, the model designed for binary classification tasks utilizes a sigmoid function, which produces a single probability value in the range between zero and one. In contrast, the model with the multiclass classification task employs a softmax activation function in the output layer, which generates a probability distribution across the multiple classes. This ensures that the sum of probabilities equals one.

In summary, while the fundamental structure of the models remains consistent, specific adaptations are made to accommodate the differences in classification tasks, including the number of classes, loss functions and output layer activation functions.

4.2 Training CNN with two Classes

The CNN model is optimized to differentiate between binary categories, here benign and malignant. Divided into two classes, the first class includes BI-RADS® classifications one, two and three, while the second includes BI-RADS® classifications four and five.

After undergoing training for 20 epochs, the outcomes of the training are visualized in figure 20. The summery in table 2 includes the loss and accuracy measurements for both the validation and test datasets. The CNN trained on benign and malignant mammography images shows a slightly promising performance on both validation and test datasets. For the evaluation, the model achieves a validation accuracy of 72.88% and a testing accuracy of 67.79%, while the baseline accuracy, representing the predictive performance of a simple model always predicting the majority class, was calculated at 66.33%. The validation and test losses, which are 2.4495 and 3.1730 respectively, indicate the level of error the model encounters while making predictions on unseen data. A higher loss value indicates that the

model's predictions differ significantly from the actual labels. The relatively high losses point out that the model struggles to accurately classify mammography images into the two classes, although it achieves appropriate accuracy rates.

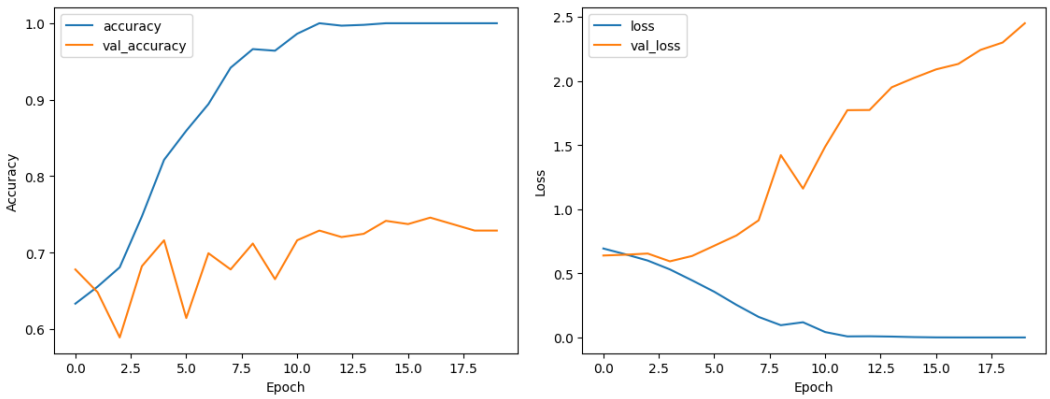

Figure 20 Training and validation performance of the binary classification model. Own illustration.

Model with two classes				
Epochs	**Validation accuracy**	**Testing accuracy**	**Validation loss**	**Testing loss**
20	0.7288	0.6779	2.4495	3.1730

Table 2 Performance of the binary classification model.

The ROC curve of the model (the orange line), which is shown in figure 21, illustrates how well it outperforms random guessing. The ROC curve should ideally be as close as possible to the upper left corner, which indicates perfect classification. The further the ROC curve is from the diagonal line, the better the performance of the model. In this instance, the ROC curve is typically situated below the diagonal line, which signifies that the model is not performing optimally. The curve suggests that the model's performance in classifying images is close to random guessing, as indicated by the AUC value of 0.51. This aligns with the relatively high losses, indicating that the model struggles to accurately classify the images. Nevertheless, the model's accuracy rates are above the baseline accuracy, indicating that it does possess some predictive power.

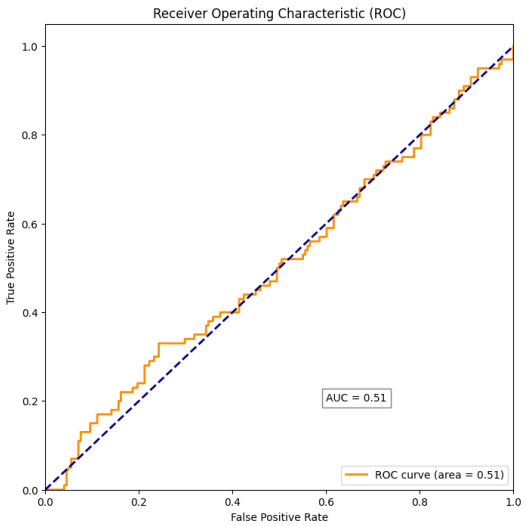

Figure 21 ROC of the binary classification model. Own illustration.

4.3 Training CNN with three Classes

The CNN model is optimized to discern between multiple categories, here normal, benign, and malignant. Divided into three classes, the normal class contains the BI-RADS® classifications one and two, the benign class contains the BI-RADS® classification three, and the malignant class contain the BI-RADS® classification four and five.

The performance of the CNN model with three classes is evaluated based on its training and validation results over 20 epochs. The outcomes of the training are visualized in figure 22. Initially, the model's accuracy steadily increases from 40.27% to 99.37%, while the loss decreases progressively. However, despite achieving perfect accuracy on the training set, the model's validation accuracy varies, reaching a peak of 56.78% at epoch 17, and then gradually decreasing. In comparison the baseline accuracy is 43.82%. Given the model’s accuracy of nearby 100% on the training dataset, the model exhibits a high level of overfitting. This occurs when a model learns specific patterns in a training dataset so well that it cannot generalize effectively to unseen datasets. While achieving perfect accuracy on the training dataset may initially appear to be a desirable outcome, it can result in poor performance on the validation or test data, thereby reducing the model's usefulness in real-world applications.

Furthermore, the summery in table 3 includes the loss and accuracy measurements for both the validation and test datasets. The model's performance on the testing dataset reveals a significant drop in accuracy, with only 47.65%

accuracy achieved. This discrepancy between training, validation, and testing accuracies suggests that the model may not generalize effectively to new, unseen data.

The observed challenges in model performance may stem from various factors, including the presence of wires, clips, and other landmarks in mammography images, which will be considered in more detail in the chapter 5. The presence of these artefacts can introduce complexity and ambiguity, potentially hindering the model's classification performance. Furthermore, the high variability in the distribution of these markers across different BI-RADS® classes further complicates the classification task. While the model demonstrates high accuracy on the training dataset, its performance on unseen data, as reflected in the testing results, falls short.

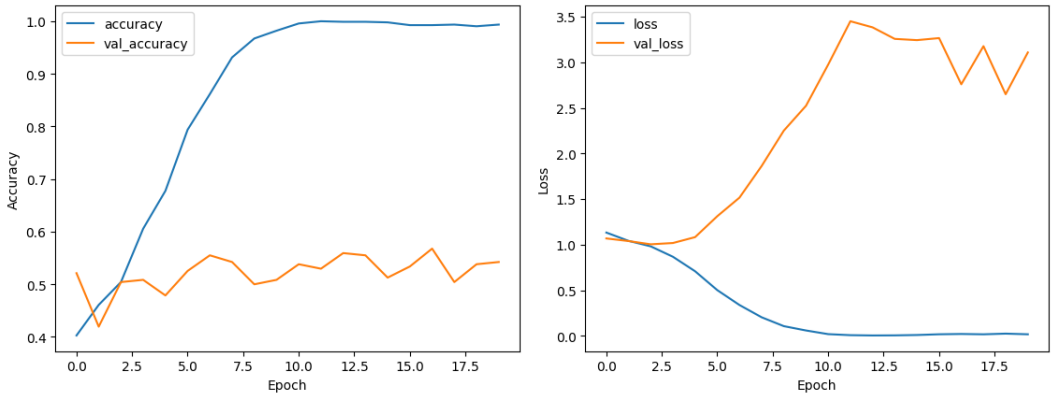

Figure 22 Training and validation performance of the multiple class performance (three classes). Own illustration.

Model with three classes				
Epochs	**Validation accuracy**	**Testing accuracy**	**Validation loss**	**Testing loss**
20	0.5424	0.4765	3.1083	3.8389

Table 3 Performance of the multiclass classification model (three classes).

Figure 23 visualizes the confusion matrix within the multiclass classification model with three classes. The matrix is a grid divided into nine cells, each representing a combination of predicted and true labels for the three classes. Each cell contains a number indicating how many instances were classified into that combination. The color scale on the right-side ranges from light blue to dark blue, representing values

from low to high. The normal class has the highest correct predictions with 93 instances correctly classified.

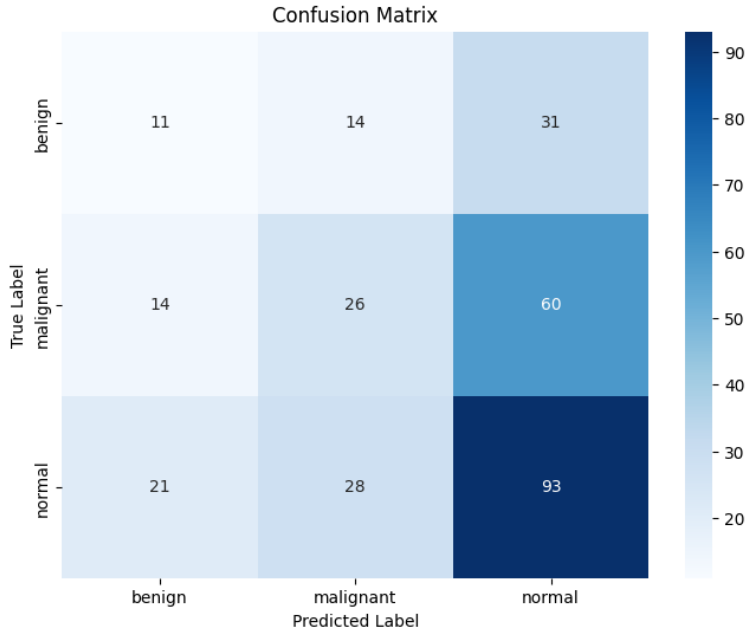

Figure 23 Confusion matrix of the multiclass classification model (three classes). Own illustration.

4.4 Training CNN with five Classes

The CNN model is adept at handling complex classification tasks, distinguishing between five categories, here the BI-RADS® classifications from one to five.

The performance of the model is interpreted and assessed based on its training and validation results over 20 epochs, as well as its testing accuracy and loss. Throughout the training process, which is visualized with the validation process in figure 24, the model demonstrates a progressive decrease in loss and an increase in accuracy on the training dataset. However, on the validation dataset, the model's performance varies, with accuracy ranging from 22.46% to 31.36% and the corresponding loss ranging from 1.5800 to 4.2117. This variation indicates potential difficulties in generalizing the learned patterns to unseen data. Compared to that the baseline accuracy is 24.70%.

Table 4 includes the loss and accuracy measurements for both the validation and test datasets. The model's performance on the testing dataset reveals a relatively low accuracy of 22.82% and a high loss of 5.8493. This discrepancy between training, validation, and testing accuracies suggests that the model may not

generalize effectively to new, unseen data, indicating potential overfitting. In conclusion, while the model demonstrates satisfactory performance on the training dataset, its effectiveness on unseen data, as reflected in the testing results, is limited.

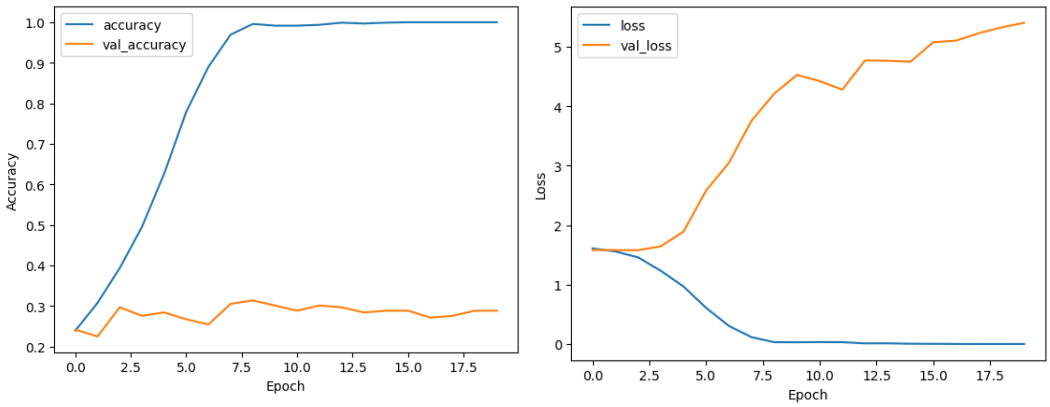

Figure 24 Training and validation performance of multiple class performance (five classes). Own illustration.

Model with five classes				
Epochs	**Validation accuracy**	**Testing accuracy**	**Validation loss**	**Testing loss**
20	0.2881	0.2282	5.4049	5.8493

Table 4 Performance of multiclass classification model (five classes).

Figure 25 visualizes the confusion matrix for the multiclass classification model with five classes. The diagonal cells from the top left to the bottom right represent correct predictions, where the predicted label matches the true label. The counts in these cells indicate the number of instances were correctly classified for each category. The off-diagonal cells indicate misclassifications, where the predicted label does not match the true label. The counts in these cells show how many instances were incorrectly classified. The confusion between BI-RADS® one and BI-RADS® two in the confusion matrix could be due to the similarities in the mammographic findings of these two categories. Both categories indicate the absence of malignancy, which implies that the mammograms within these categories are generally normal. Nevertheless, the differentiation between these two categories lies in the specific findings mentioned in the mammogram report. BI-RADS® one is assigned when no specific benign findings are mentioned in the report, even if they might be present. Conversely, BI-RADS® two is assigned when

one or more specific benign mammographic findings are mentioned in the report, including calcified fibroadenomas, skin calcifications, metallic foreign bodies, or fat-containing lesions. It is possible that the model may encounter difficulties in distinguishing between these two categories, particularly in cases where benign findings are present but not explicitly mentioned in the report. This could result in a higher number of instances where the model predicts BI-RADS® two when the true label is BI-RADS® one, or vice versa, which would lead to higher numbers in the confusion matrix for these two categories.

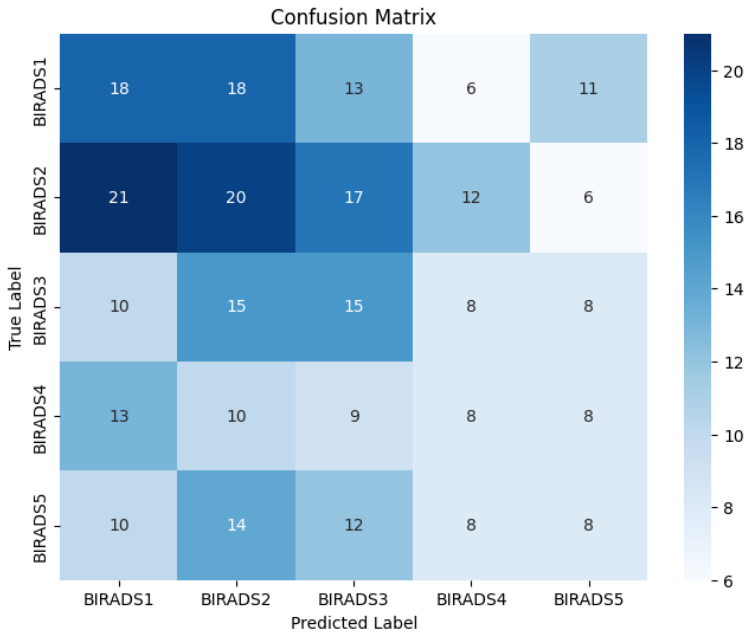

Figure 25 Confusion matrix of the multiclass classification model (five classes). Own illustration.

4.5 Enhancing Model Performance

The following section will examine methods for fine-tuning CNNs in order to improve their accuracy and reliability. These methods include the implementation of additional layers and data augmentation.

4.5.1 Additional Layers

In order to enhance the efficiency of the model, it was augmented with convolutional layers, as illustrated in listing 21.

```
    layers.Conv2D(128, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
```

```
    layers.Conv2D(256, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
```

Listing 21 Additional convolutional layers.

As illustrated in table 5, the binary classification model with additional layers shows a marginal enhancement in testing accuracy (0.7013 versus 0.6779) and a reduction in testing loss (1.2656 versus 3.1730). Nevertheless, the improvement is not substantial, and the validation accuracy has decreased slightly. It is essential to consider whether the additional complexity justifies this marginal improvement. As in the binary case, the additional layers in the multiclass classification model with three classes do not significantly improve performance. In fact, the accuracy of the testing process appears to decline, suggesting that the additional complexity may not be beneficial for this multiclass problem. Similarly, for the multiclass classification model with five classes, the additional layers do not result in a notable enhancement. The testing accuracy remains low, and the losses are high. The additional convolutional layers appear to have a limited impact on the model's performance across all three scenarios.

Model performance – Additional Layers					
Classifi-cation	**Addit. Layers**	**Validation accuracy**	**Testing accuracy**	**Validation loss**	**Testing loss**
Binary	No	0.7288	0.6779	2.4495	3.1730
	Yes	0.6949	0.7013	1.1496	1.2656
Multiclass (three)	No	0.5424	0.4765	3.1083	3.8389
	Yes	0.5169	0.4161	2.9405	3.3694
Multiclass (five)	No	0.2881	0.2282	5.4049	5.8493
	Yes	0.3093	0.2718	6.3895	6.5704

Table 5 Model performance with and without additional layers.

As described by Abdelhafiz et al. (2019), the model can extract additional features by incorporating additional layers. However, this is only possible to a certain extent with a limit. Instead of extracting features, the additional layers result in overfitting of the network beyond this limit, which may conduct in false positive results. In the case of large datasets, the addition of further hidden layers increases the accuracy. However, if a small dataset is used, the accuracy of the test data is reduced by the

number of increased parameters due to the additional layers in the CNN. A compromise between the accuracy of the model and the depth of the network must be identified based on the dataset through a process of trial and error.

4.5.2 Data Augmentation

In order to enhance the efficacy of the models, the technique of data augmentation was also employed as described in chapter 3.5.2, in addition to the additional layers. The accuracy and loss of the models changed as outlined in the table 6.

Model performance – Data Augmentation					
Classifi-cation	**Data Augm.**	**Validation accuracy**	**Testing accuracy**	**Validation loss**	**Testing loss**
Binary	No	0.7288	0.6779	2.4495	3.1730
	Yes	0.6737	0.6140	0.6643	0.6921
Multiclass (three)	No	0.5424	0.4765	3.1083	3.8389
	Yes	0.5381	0.4195	1.0514	1.1635
Multiclass (five)	No	0.2881	0.2282	5.4049	5.8493
	Yes	0.2797	0.2081	1.6212	1.7724

Table 6 Model performance with and without applying data augmentation.

For the binary classification model data augmentation slightly reduces the validation accuracy (0.6737 versus 0.7288) but improves the testing accuracy (0.6140 versus 0.6779). The lower validation loss (0.6643 versus 2.4495) suggests better generalization. Overall, data augmentation seems beneficial for the binary classification task, as it better helps the model generalizing to unseen data. Nevertheless, the impact of data augmentation on the performance of the multiclass classification model with three classes is not statistically significant. The testing accuracy decreases slightly, but the validation loss improves. Similarly, for the multiclass classification model with five classes, data augmentation has a minimal impact. The testing accuracy decreases, but the validation loss improves.

5 Discussion

According to Géron (2019) machine learning is a generally useful technique for addressing a diverse array of problems and applications. Applying machine learning techniques to analyze large amounts of data can help to uncover patterns that may not be immediately apparent – this is known as data mining. For challenges that demand thorough adjustments or extensive sets of instructions, employing a single machine learning algorithm often simplifies the code and outperforms conventional methods. Additionally, machine learning techniques offer solutions for complex issues that are beyond the capabilities of traditional approaches. Furthermore, machine learning systems can adapt to new data and changing circumstances, providing insights into multifaceted problems and extensive datasets.

In accordance with the findings of Isosalo et al. (2023), it is not uncommon for breasts that are considered to be within the normal range to exhibit certain changes when examined using mammography. These changes might involve specific occurrences like calcifications and steatonecrosis, also known as fat tissue necrosis. Even findings that are clearly benign, such as fibroadenomas and calcifications that have remained unchanged for an extended period, could be categorized as "normal" during screenings. Some benign discoveries, like well-defined masses with larger calcifications, masses showing fatty or mixed density, and calcifications within the arteries (known as vascular breast calcifications), display distinct non-cancerous characteristics. However, benign entities such as cysts and lymph nodes possess visual features that can be mistaken for malignancy. As a result, interpreting breast patterns can pose challenges when training an effective breast cancer classifier. Non-specific findings often necessitate further evaluation using additional methods like ultrasound.

5.1 Evaluation of the Mammography Image Dataset

In order to undertake a critical evaluation of the dataset, it is necessary to pose the following question. If a bilateral mammogram was performed, i.e. both breasts of a person – left and right, each in MLO and CC projection – were imaged, and a joint report was written for both breasts, in which, for example, a BI-RADS® classification of five was noted, does this BI-RADS® classification apply to both

breasts? Or would one side of the breast possibly have a different BI-RADS® classification? If the breasts were examined independently, would one breast be in a different BI-RADS® folder compared to the other breast? For example, if there is a suspicious center, the findings will show a BI-RADS® five on the right breast. Meaning, that there is a BI-RADS® classification in the report, but usually only one side is affected, and the report should state which side that BI-RADS® classification applies to. If both sides of the breast are examined and a joint report is made, the report may include either a BI-RADS® classification for each side of the breast or only the higher BI-RADS® classification. As the data for this thesis was provided by the Ordensklinikum Linz GmbH Barmherzige Schwestern as a list with the patient data and the corresponding BI-RADS® classification, it was not clear when extracting the data from the PACS whether the BI-RADS® classification specified in the list applied to only one side of the breast or to both sides of the breast in the case of a bilateral examination.

A potential challenge and reason for low model accuracy can be the presence of wires, clips and other landmarks in mammography images. Especially in mammography images of patients with biopsies or previous surgery. These landmarks, clips, etc., which are visible in different BI-RADS® classes, can add complexity and ambiguity to the classification. The variability in the distribution of these markers across different BI-RADS® classes potentially further complicates the classification task. As the Ordensklinikum Linz GmbH Barmherzige Schwestern is a leading oncology hospital, there is a high frequency of mammograms with previous surgical procedures, wires, clips, etc. among its patient clientele. Clips from previous breast surgeries, mammograms with wire markings in preparation for surgery, biopsies, or multiple surgeries, etc. This variability of external factors in the mammography images and between the BI-RADS® classes, as shown in figure 26, may also complicate the model's classification of the medical image data, which may in turn explain its low accuracy.

As Abdelhafiz et al. (2019) observed, training a CNN model with a limited amount of medical data, as is the case in this thesis, represents a significant challenge. According to Abdelhafiz et al. (2019) the associated problems can be mitigated by the utilization of transfer learning and augmentation techniques. Moreover, research has demonstrated that CNN models analyzing images from both left and right breasts, along with views from both the CC and MLO perspectives of each breast, can enhance the detection accuracy and minimize the number of false positives.

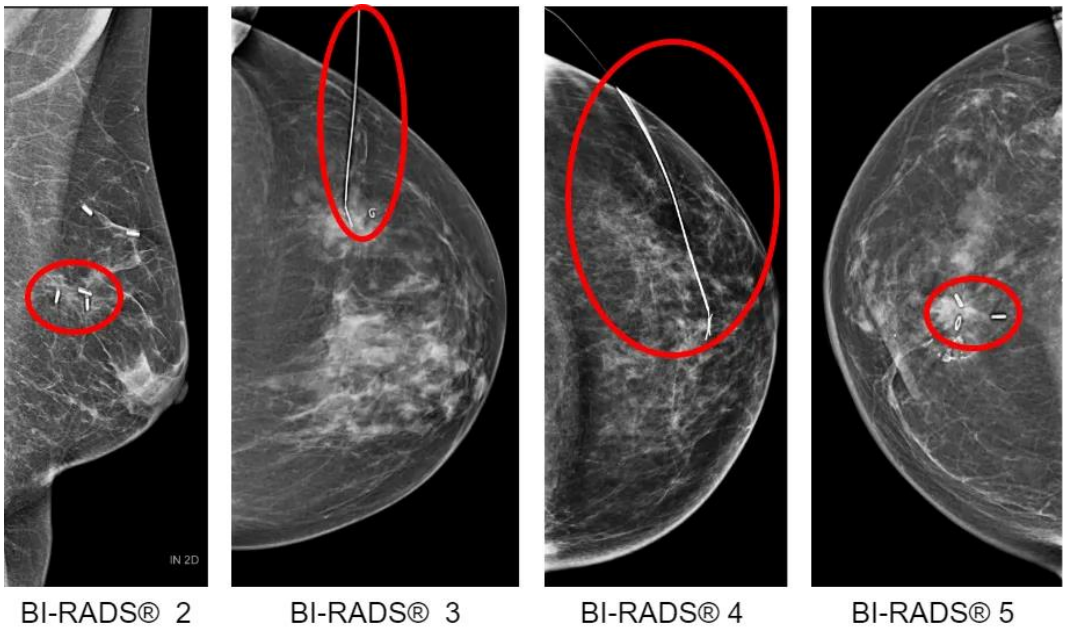

Figure 26 Mammography images with surgical landmarks and artifacts. Images of BIRADS two and five show clips. Images of BI-RADS® three and four show wire markings. Own illustration based on Ordensklinikum Linz GmbH Barmherzige Schwestern, 2023.

5.2 CC and MLO Projections

According to Sridevi & Samath (2024) in conventional CAD systems for breast cancer detection, the CC and MLO projections are typically used independently. Following the training phase, predictions are generated for each view individually. Recently there has been a growing interest in the integration of information from both CC and MLO views, with the objective of enhancing the overall accuracy.

A true anomaly is typically identified in two distinct perspectives of a mammography image as mentioned by Abdelhafiz et al. (2019). The application of multi-view methodologies has led to significant improvements in comparison to single-view approaches, as evidenced by recent research. The comparison of two perspectives can contribute to the reduction of false positive and false negative results.

The incorporation of both projections enables the CNN to utilize complementary information from diverse angles, thereby enhancing its overall performance (Baswaraju, 2023).

One such approach is feature fusion, as outlined by Sridevi & Samath (2024), whereby the extracted features from both views are merged prior to classification.

Sridevi & Samath (2024) proposed a technique that merges a CNN with feature fusion using the least absolute shrinkage and selection operator (LASSO) regression. The extracted features from both CC and MLO projections are obtained from pre-trained CNN models. Afterwards, the aforementioned features from the two different views are merged together through the use of LASSO regression. The fused features are subsequently input for the fully connected layer of the CNN for the purpose of classifying mammograms. The objective of this integrated approach is to utilize complementary information from both perspectives, with the potential to enhance accuracy.

Baswaraju (2023) mentioned that by considering multiple perspectives, the model can reduce the number of false positives, thereby improving the precision of the predictions. The integration of CC and MLO projections enhances the sensitivity of anomaly detection. However, it is important to note that the precise alignment of CC and MLO images is crucial in view of the differences in image positioning.

Conclusively, although traditional methods, as used in this thesis, have focused on individual approaches for CC and MLO projections, Sridevi & Samath (2024) have indicated that the integration of features derived from both perspectives may enhance the precision of breast cancer detection in CAD systems. Abdelhafiz et al. (2019) outlined that the utilization of multi-view approaches has been demonstrated to yield considerable enhancements in comparison to single-view approaches. Furthermore, the comparison of two projections can assist in minimizing false positives and false negatives.

A further step beyond the usage of two different views would be to utilize multimodal machine learning as explained by Abdelhafiz et al. (2019). The recognition rate of CNNs can be enhanced not only by employing different projections (CC or MLO), but also by incorporating information such as the patient's age or breast density. Multimodal machine learning is a field of study that aims to create models capable of processing and relating information from multiple modalities, such as images and text, combining the information with a score level in the final predictions.

5.3 Data Augmentation

Data augmentation may not be suitable for mammography image classification for several reasons. Mammography images capture the breast typically in the MLO and CC projections. Augmenting these images with rotations or reflections can result in unrealistic variations that do not reflect actual breast anatomy. Introducing

random rotations or zooms can distort anatomical features that are critical to the diagnosis. This could lead to misinterpretation of the images and inaccurate classification by the model. Introducing artificial variations through augmentation can distort critical image features for breast cancer screening or diagnosis, potentially leading to false diagnoses.

Although data augmentation is a common technique in image classification tasks, it may be prudent to use augmentation techniques with caution, if at all, when dealing with mammography images. In addition, considering the importance of grey level intensity information in mammography, all pre-processing steps should preserve the grey level information as natural as possible.

5.4 Transfer Learning

As explained by Géron (2019), transfer learning involves applying an already trained model to a new but similar task, rather than training a new model from scratch. With the help of transfer learning, low-level structures of already pre-trained models are used and only the structures on higher layers must be learned. Rather than randomly initializing the weights and biases of the first layers of the new model, they are initialized with the values of the lower layers of the previous model. This approach allows the new model to focus on learning higher-level features without the need to relearn all the basic features. It is recommended to search for already existing neural networks that address a comparable task to the one being pursued.

According to (Abdelhafiz et al., 2019), in the field of medicine, there is a necessity for transfer learning due to the limited availability and high cost of data, its lack of public accessibility, and the time-consuming process of collecting and annotating it from medical specialists. Frequently used pre-trained networks for medical image classification are Alex-Net, ZF-Net, GoogLeNet, VGG-Net and ResNet.

The benefits of transfer learning include not only significantly speeding up the training process but also a reduction in the quantity of training data required, as outlined by Géron (2019). It is important to consider if the input images for the new task differ in size from those used in the original task. Moreover, it is necessary to preprocess the pictures to adjust their dimensions to match the required size in the original model. The benefit of transfer learning can be demonstrated using a scenario where a pre-trained deep neural network (DNN) is used to classify images into 100 different categories. The goal is to train another DNN specifically tailored to classify specific types of vehicles. Due to the similarity of these tasks, it is

recommended to use parts of the existing network as visualized in figure 27. Typically, the output layer of the original model needs to be replaced as it may not be suitable for the new task and may not have the required number of outputs. Similarly, the upper hidden layers of the original model are expected to be less useful compared to the lower layers. This is because there may be significant differences between the high-level features relevant to the new task and the features preferred for the original task.

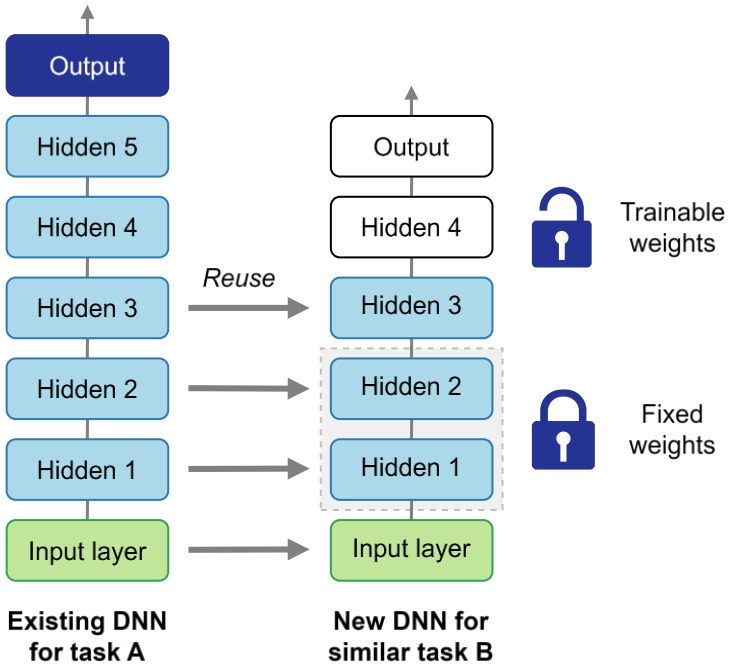

Figure 27 Reusing pretrained layers. Own illustration based on Géron, 2019, p. 453.

Transfer learning is less effective on small and dense networks as explained by Géron (2019). This is primarily due to the limited pattern diversity learned by small networks and the highly specific patterns acquired by dense networks, which may not be useful for alternate tasks. CNNs are best suited for transfer learning due to their tendency to acquire more generalized feature detectors.

One challenge in applying transfer learning to a CNN for mammography image classification is the variability in the dimensions of mammography images. Mammography images are typically acquired in different orientations (MLO, CC), resulting in variations in image dimensions. To facilitate model training, these images need to be resized to a consistent size. However, resizing mammography images to a square dimension, as commonly used in transfer learning models, requires the addition of additional black space to compensate for the differences in dimension ratio. This additional black space raises several concerns. It changes the original image content, which can introduce noise into the data. And this

change can impact the model's ability to accurately classify anomalies, especially ones that rely on fine detail. In addition, the presence for black space can affect the performance of the model, as it learns from both relevant image features and irrelevant background information. Furthermore, the addition of black space can increase the data sparsity problem that is often present in medical image datasets. Since there are few annotated mammography images available for training, the inclusion of black space artificially increases the size of the dataset without adding meaningful information. This could lead to overfitting, where the model learns to detect irrelevant patterns associated with the black space, rather than focusing on clinically relevant features.

However, after the images were brought to the same dimension ratio, even without the use of transfer learning, additional black space had to be added. This phenomenon may also contribute to the relatively low accuracy of the models described in chapter four.

Although transfer learning has the potential to improve CNN performance in mammography image classification, the need for image adjustment displays a challenge. The introduction of black space during resizing can affect both the quality of the data as well as the effectiveness of the model.

6 Conclusion

This thesis aimed to develop and evaluate a CNN for the automatic classification of mammography images. The models were trained and evaluated using two, three and five classes of mammography images, with the test data representing 20% of the total image data to be trained. The performances of the models were evaluated by calculating the accuracy, the loss function, the ROC, and the confusion matrix. The results provide insight into the challenges associated with training a CNN for mammography image classification. Despite the considerable investment of resources in model development and optimization, the results demonstrated variable performance. While the CNN exhibited promising levels of accuracy on the training data, its capacity to generalize to unseen data, such as that of the test dataset, was limited.

While existing models available on the internet or through companies are convenient, building a CNN from scratch offers scientific value and benefits as well as several advantages. The construction of a CNN from the ground up enables the architectural and parameter customization to the specific conditions and complexities of mammography image analysis. Furthermore, the construction of a CNN from scratch facilitates a more profound comprehension of the fundamental principles and design of neural networks and image classification. The provision of comprehensive documentation regarding the model architecture, training process, and evaluation metrics serves to enhance the reproducibility of the research. This transparency enables the results to be reproduced and validated.

Although the model developed in this thesis did not fulfil the requirements of high accuracy and good performance, and is therefore not suitable for clinical practice, the literature indicates that there is a considerable potential for CAD systems in the clinical diagnostic field.

Jairam & Ha (2022) posits that the usage of artificial intelligence has the potential to facilitate the early detection of a small tumor before it is visible for radiologist. Furthermore, the deployment of CAD systems has the capacity to enhance the efficiency of radiologists by identifying and prioritizing likely negative mammograms, thereby enabling medical personnel to focus on interpreting abnormal mammograms. This, in turn, has the potential to reduce the workload of medical personnel and the associated costs of screening programs. In addition to the previously mentioned components, other aspects also pose a challenge. These include economic aspects, such as the costs of implementation, ethical and legal

aspects, such as consent, liability, data protection, and cybersecurity. The potential of artificial intelligence-based technologies in the diagnostic field, and specifically in the medical field of mammography, is considerable. Despite the challenges, according to Jairam & Ha (2022), artificial intelligence will improve as an aid in medical diagnostics and continue to offer an improvement in breast cancer screening.

In conclusion, while the utilization of pre-existing CNN models may be expedient, the scientific advantage and added value of developing a CNN from scratch for the automatic classification of mammography images lies in its specific design, the deeper understanding of neural network principles, and the promotion of research reproducibility. This approach enables a meaningful contribution to the advancement of breast cancer diagnosis and personalized medicine.

References

Abdelhafiz, D., Yang, C., Ammar, R., & Nabavi, S. (2019). Deep convolutional neural networks for mammography: Advances, challenges and applications. *BMC Bioinformatics*, *20*(S11), 281. https://doi.org/10.1186/s12859-019-2823-4

Abdelrahman, L., Al Ghamdi, M., Collado-Mesa, F., & Abdel-Mottaleb, M. (2021). Convolutional neural networks for breast cancer detection in mammography: A survey. *Computers in Biology and Medicine*, *131*, 104248. https://doi.org/10.1016/j.compbiomed.2021.104248

American College of Radiology, D'Orsi, C. J., Sickles, E. A., Mendelson, E. B., & Morris, E. A. (Eds.). (2013). *ACR BI-RADS atlas: Breast imaging reporting and data system ; mammography, ultrasound, magnetic resonance imaging, follow-up and outcome monitoring, data dictionary* (5th edition). ACR, American College of Radiology.

Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An Introduction to Machine Learning. *Clinical Pharmacology & Therapeutics*, *107*(4), 871–885. https://doi.org/10.1002/cpt.1796

Baswaraju, S. (2023). Enhancing Breast Cancer Classification in Mammography Images Using Multi-View Deep Convolutional Neural Networks. *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, 1572–1577. https://doi.org/10.1109/ICOSEC58147.2023.10276164

Bill, R., & Berger, M. (n.d.). *Tutorial: Datenformate*. Universität Rostock - Agrar- und Umweltwissenschaftliche Fakultät. https://learn.opengeoedu.de/tutorials/OGE-Tutorial_Dateiformate.pdf

Binny, M., & Omair, A. (2024, February 6). Multiclass Confusion Matrix—All That You Need to Know. *ProjectPro*. https://www.projectpro.io/recipes/explain-multiclass-confusion-matrix

Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., & Jemal, A. (2018). Global cancer statistics 2018: GLOBOCAN estimates of incidence

and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, *68*(6), 394–424. https://doi.org/10.3322/caac.21492

Bunker, R. P., & Thabtah, F. (2019). A machine learning framework for sport result prediction. *Applied Computing and Informatics*, *15*(1), 27–33. https://doi.org/10.1016/j.aci.2017.09.005

Carneiro, G., Nascimento, J., & Bradley, A. P. (2015). Unregistered Multiview Mammogram Analysis with Pre-trained Deep Learning Models. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (Vol. 9351, pp. 652–660). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_78

Clark, J. A. (2024). *Pillow (PIL Fork) 10.2.0 documentation* [Documentation]. Pillow. https://pillow.readthedocs.io/en/stable/

Deutsches Krebsforschungszentrum. (2023, June 15). *Brustkrebs (Mammakarzinom)* [Krebsinformationsdienst]. https://www.krebsinformationsdienst.de/tumorarten/brustkrebs/index.php

Geras, K. J., Wolfson, S., Shen, Y., Wu, N., Kim, S. G., Kim, E., Heacock, L., Parikh, U., Moy, L., & Cho, K. (2018). *High-Resolution Breast Cancer Screening with Multi-View Deep Convolutional Neural Networks* (arXiv:1703.07047). arXiv. http://arxiv.org/abs/1703.07047

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow—Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.). O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472e.

Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). Fundamental Concepts of Convolutional Neural Network. In V. E. Balas, R. Kumar, & R. Srivastava (Eds.), *Recent Trends and Advances in Artificial Intelligence and Internet of Things* (Vol. 172, pp. 519–567). Springer International Publishing. https://doi.org/10.1007/978-3-030-32644-9_36

Isosalo, A., Inkinen, S. I., Turunen, T., Ipatti, P. S., Reponen, J., & Nieminen, M. T. (2023). Independent evaluation of a multi-view multi-task convolutional neural network breast cancer classification model using Finnish mammography screening data. *Computers in Biology and Medicine*, *161*, 107023. https://doi.org/10.1016/j.compbiomed.2023.107023

Jairam, M. P., & Ha, R. (2022). A review of artificial intelligence in mammography. *Clinical Imaging*, *88*, 36–44. https://doi.org/10.1016/j.clinimag.2022.05.005

Kumar, T., & Verma, K. (2010). A Theory Based on Conversion of RGB image to Gray image. *International Journal of Computer Applications*, *7*(2), 5–12. https://doi.org/10.5120/1140-1493

Kundu, R. (2022, September 13). Confusion Matrix: How To Use It & Interpret Results [Examples]. *Machine Learning*. https://www.v7labs.com/blog/confusion-matrix-guide

Lee, A. (2021, May 7). Choosing a Baseline Accuracy for a Classification Model Pick a simple baseline accuracy to demonstrate that your classification model has skill on a problem [A Medium publication sharing concepts, ideas and codes for data science.]. *Towards Data Science*. https://towardsdatascience.com/calculating-a-baseline-accuracy-for-a-classification-model-a4b342ceb88f

Mohi ud din, N., Dar, R. A., Rasool, M., & Assad, A. (2022). Breast cancer detection using deep learning: Datasets, methods, and challenges ahead. *Computers in Biology and Medicine*, *149*, 106073. https://doi.org/10.1016/j.compbiomed.2022.106073

Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python—A Guide for Data Scientists* (1st ed.). O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Narkhede, S. (2018, June 26). Understanding AUC - ROC Curve [A Medium publication sharing concepts, ideas and codes for data science.]. *Towards Data Science*. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Ordensklinikum Linz GmbH Barmherzige Schwestern. (2023). *Mammography Images* (Version 1) [DICOM].

Österreichische Gesundheitskasse. (2023, October 3). Früh erkennen, Österreichisches Brustkrebs-Früherkennungsprogramm [Die Österreichische Gesundheitskasse betreibt diese Internetseiten zur Verbesserung der Information über ihre Aufgabenbereiche und Servicedienste.]. *früh erkennen*. https://www.frueh-erkennen.at/

Ragab, D. A., Attallah, O., Sharkas, M., Ren, J., & Marshall, S. (2021). A framework for breast cancer classification using Multi-DCNNs. *Computers in Biology*

and Medicine, *131*, 104245. https://doi.org/10.1016/j.compbiomed.2021.104245

Ramasubramanian, K., & Singh, A. (2019). Deep Learning Using Keras and TensorFlow. In K. Ramasubramanian & A. Singh, *Machine Learning Using R* (pp. 667–688). Apress. https://doi.org/10.1007/978-1-4842-4215-5_11

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Shan, H., Vimieiro, R. B., Borges, L. R., Vieira, M. A. C., & Wang, G. (2023). Impact of loss functions on the performance of a deep neural network designed to restore low-dose digital mammography. *Artificial Intelligence in Medicine*, *142*, 102555. https://doi.org/10.1016/j.artmed.2023.102555

Smirnov, E. A., Timoshenko, D. M., & Andrianov, S. N. (2014). Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks. *AASRI Procedia*, *6*, 89–94. https://doi.org/10.1016/j.aasri.2014.05.013

Songsaeng, C., Woodtichartpreecha, P., & Chaichulee, S. (2021). Multi-Scale Convolutional Neural Networks for Classification of Digital Mammograms With Breast Calcifications. *IEEE Access*, *9*, 114741–114753. https://doi.org/10.1109/ACCESS.2021.3104627

Sridevi, V., & Samath, J. A. (2024). A combined deep CNN-lasso regression feature fusion and classification of MLO and CC view mammogram image. *International Journal of System Assurance Engineering and Management*, *15*(1), 553–563. https://doi.org/10.1007/s13198-023-01871-x

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, *71*(3), 209–249. https://doi.org/10.3322/caac.21660

Suresh, A. (2020, November 17). What is a confusion matrix? [Analytics Vidhya is a community of Analytics and Data Science professionals.]. *Analytics Vidhya*. https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

Vaidya, V., Rajan, P., & Chavez, T. (2024, March 13). Why Establish Baseline Models | A Detailed Guide. *Machine Learning*. https://www.markovml.com/blog/baseline-models

Wang, J., Ding, H., Bidgoli, F. A., Zhou, B., Iribarren, C., Molloi, S., & Baldi, P. (2017). Detecting Cardiovascular Disease from Mammograms With Deep Learning. *IEEE Transactions on Medical Imaging*, *36*(5), 1172–1181. https://doi.org/10.1109/TMI.2017.2655486

Weissensteiner, S. (2012). PACS – Grundlagen, Fehlermanagement und Administration. *Radiopraxis*, *5*(03), 151–160. https://doi.org/10.1055/s-0032-1309969

Wessel, M., & Wyant, T. (2021, November 8). Breast Cancer Stages [American Cancer Society]. *Understanding a Breast Cancer Diagnosis*. https://www.cancer.org/cancer/types/breast-cancer/understanding-a-breast-cancer-diagnosis/stages-of-breast-cancer.html

Wild, C. P., Weiderpass, E., & Stewart, B. W. (2020). *World Cancer Report: Cancer Research for Cancer Prevention*. International Agency for Research on Cancer (IACR) - World Health Organization (WHO). https://publications.iarc.fr/Non-Series-Publications/World-Cancer-Reports/World-Cancer-Report-Cancer-Research-For-Cancer-Prevention-2020

Wuttke, L. (n.d.). Künstliche Neuronale Netzwerke: Definition, Einführung, Arten und Funktion. *datasolut - Mehr Wert mit KI.* Retrieved 3 March 2024, from https://datasolut.com/neuronale-netzwerke-einfuehrung

List of Figures

List of Tables

List of Listings

Conceptual definition

Adam ..... *Adaptive Moment Estimation*

ANN ..... *Artificial Neural Network*

API ..... *Application Programming Interface*

AUC ..... *Area Under the Curve*

AUROC ..... *Area Under the Receiver Operating Characteristic*

BI-RADS® ..... *Breast Imaging-Reporting and Data System*

CAD ..... *Computer aided detection*

CC ..... *Bilateral craniocaudal*

CNN ..... *Convolutional Neural Network*

CT ..... *Computer Tomography*

DICOM ..... *Digital Imaging and Communication in Medicine*

DNN ..... *Deep Neural Network*

FN ..... *False Negative*

FP ..... *False Positive*

FPR ..... *False Positive Rate*

GPU ..... *Graphics Processing Unit*

IACR ..... *International Agency for Research on Cancer*

ID ..... *Identification*

IEEE ..... *Institute of Electrical and Electronics Engineers*

LASSO ..... *Least Absolute Shrinkage and Selection Operator*

MLO ..... *Mediolateral oblique*

MLPs ..... *Multilayer perceptrons*

MRI ..... *Magnetic Resonance Imaging*

PACS ..... *Picture Archiving and Communication System*

PET ..... *Positron Emission Tomography*

PIL ..... *Python Imaging Library*

PNG ..... *Portable Network Graphics*

ReLU ..... *Rectified Linear Unit*

RMSprop ..... *Root Mean Square Propagation*

ROC ..... *Receiver Operating Characteristic*

TN ..... *True Negative*

TNR ..... *True Negative Rate*

TP ..... *True Positive*

TPR ..... *True Positive Rate*

uint ..... *unsigned integer*

WHO ..... *World Health Organization*

ZeroR ..... *Zero Rate Classifier*