Uncertainty Quantification in Deep Learning for Breast Cancer Classification in Point-of-Care Ultrasound Imaging

Marisa Wodrich

LUND
UNIVERSITY

Department of Mathematics

Abstract

Breast cancer is the most common type of cancer worldwide with an estimate of 2.3 million new cases in 2020, and the number one cause of cancer-related deaths in women. While survival rates are high in many high-income countries, with a five year relative survival rate of 85% and more, the respective survival rates are poor in many middle- and low-income countries, with rates as low as 12% in Kyadondo, Uganda. This immense difference is largely due to the difference in availability of access to diagnostic tools and screenings, as well as the amount of diagnostic experts.

One solution to bridge this gap and increase the survival rates in low-income countries could be to use point-of-care ultrasound (POCUS) imaging as a cheap and portable diagnostic tool, combined with a deep learning (DL) based algorithm for image classification. While it has previously been shown that this is possible and can produce good results, it is extremely important in a field like medical diagnostics to have a classifier that is also trustworthy, as wrong predictions can have severe consequences.

This work therefore addresses the question of how to quantify uncertainties in a model's prediction and explores different methods from the field of uncertainty quantification (UQ) and out-of-distribution (OOD) detection, including Bayesian neural networks, deep ensembles and three different post-hoc methods. The results support the hypothesis that there is a correlation between uncertainty scores and the correctness of a prediction. The correlation was the strongest using an average ensemble with entropy-based total uncertainty. The results suggest that a suitable threshold should be set so that the predictions of the 20% of test data with the highest uncertainties will be marked as not trustworthy. This improves the accuracy of the breast cancer classification (benign, malignant, normal) from previous 68.6% to 77.5%, binary accuracy (cancerous vs. non-cancerous) from 81.8% to 90.2%, and the AUC from 95.6% to 98.4%.

Additionally, all methods were tested for the purpose of OOD detection using three different OOD data sets. The best results were achieved using the post-hoc OOD detection method energy score, performing well on all three data sets, followed by several types of ensembles.

Overall, the results show that there is great potential in the different methods for the purpose of building a safer and more trustworthy classifier that can be applied in a real-world setting. Based on our findings, an average ensemble as the classification method with entropy-based total uncertainty is the most promising choice, followed by the energy score method. Further evaluation with more data and comparison to additional UQ methods is needed to confirm the optimal method.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Ida Arvidsson and Jennie Karlsson for their invaluable support and guidance during this project. I appreciate the time they took to help me and their willingness to impart their knowledge. I would like to extend my sincere thanks to Kristina Lång for her help with collecting data at the hospital and taking her time explaining ultrasound images to me. It has been a true inspiration working with so many inspiring, smart and encouraging women who taught me a lot.

Furthermore, I would like to thank everyone at the research group for the great work atmosphere, making me feel so welcome, and their generous support during the time of this project. Your excitement and passion for research has truly inspired me. Lastly, I would like to thank my amazing friends and family for their emotional support. Their belief in me has encouraged me to explore my research interests and has kept my spirits and motivation high during this process.

Contents

1

Introduction

1.1 Motivation

Breast cancer is the most common type of cancer worldwide and the number one cause of cancer related deaths in women [1]. In order to reduce breast cancer mortality and morbidity, large screening programs using mammography have been implemented in many countries. This has shown promising results, having led to a significant reduction in mortality rates [2, 3]. However, due to the high cost of mammography machines and the intense training of medical personnel needed to use and interpret mammography, it is unfeasible to implement such a solution in practice in many low-income countries.

Karlsson et al. [4] have proposed a novel approach with promising results to possibly replace mammography screenings in such regions, which uses point-of-care ultrasound (POCUS) imaging and a convolutional neural network (CNN) to classify breast lesions. Using POCUS imaging instead of mammography solves the problem of cost, while also being applicable in settings with poor health infrastructure, due to POCUS devices being cheap, small and portable. Furthermore, they are easy to use with short training. Using a CNN to evaluate the images in real time additionally eases the education that the health personnel would have to undergo. The findings in [4] serve as a baseline for this work.

In recent years there has been an increased interest in using deep learning (DL) for medical applications. DL has shown to produce high-quality results in assessing different types of medical images [5]. While DL models generally perform well on these tasks, achieving high accuracies for many applications, they tend to make overconfident decisions. This can be caused by their weak capabilities of quantifying predictive uncertainties [6]. Overconfident predictions can lead to errors which can be both harmful and offensive [7]. To ensure the safety and trustworthiness of a neural network, it is crucial that it has the ability to report how certain it is at its decision and when it cannot make a reliable assessment. This becomes especially important in critical domains like medical diagnostics, where wrong predictions can have severe consequences. Therefore, suitable measurements for uncertainty have to be found and should be key features in medical DL applications [8].

The field of uncertainty quantification (UQ) aims to find methods that can determine how certain a network is about its prediction for a certain input. Some of these methods are novel approaches on how to design the network, others are applied post-hoc, meaning after making a prediction with a model. An ideal classifier would perform well at assessing images for which its uncertainty is low, and possibly perform worse on images with high predictive uncertainties. Furthermore, a similar concept holds for out-of-distribution (OOD) detection, implying that UQ and OOD detection methods should be able to flag data samples which are outside of the distribution of the training data. Since the network has not been trained to process and interpret such data sample, the prediction should also not be trusted in this case.

1.2 Aim

The aim of this work is to extend the previous findings from Karlsson et al. [4] by investigating how UQ and OOD detection methods can be used to improve the classifier's safety and trustworthiness, making it more applicable in a real-world setting. In the scope of this work five different UQ and OOD detection methods have been implemented and compared for the application of breast cancer classification in POCUS imaging. The respective methods are Bayesian neural networks (BNNs), neural network ensembles, and the post-hoc OOD detection methods softmax output, trust score and energy score. Several types of Bayesian networks and ensembles have been evaluated, leading to a total of 20 different UQ methods. All methods are evaluated for the purpose of quantifying predictive uncertainties for in-distribution POCUS data, as well as for their performance at OOD detection on three different OOD data sets.

This work is split up into three consecutive studies. The first study compares the classification performance of the networks underlying the different UQ methods. The second study compares the different methods for the task of quantifying predictive uncertainties in the test data, with the goal to find an uncertainty threshold that can detect samples that are likely wrongly classified and should not be trusted. The last experiment aims at using the UQ methods to detect OOD data from three different OOD data sets.

2

Background

2.1 Point-of-Care Ultrasound

POCUS devices are small, low-cost ultrasound devices that are considerably more portable and compact than traditional, conventional ultrasound (US) systems. Since their first employment in 1998 [9], they have been used in a variety of different medical applications, including emergency care, critical care, vascular medicine, obstetrics, cardiology and rheumatology [10, 11]. POCUS has shown to have great potential in settings where resources are limited [12, 13] and thus is a promising diagnostic tool to advance global health.

Many POCUS frameworks come with examination-specific presets for several different applications, including breast imaging, making them usable in a wide range of applications. The presets define, amongst others, the depth and frequency at which POCUS is performed, which can be tissue- or organ-specific. Similar to convential US, several POCUS systems have options for advanced imagining tools like full-spectrum Doppler, M-mode and linear measurements, as well as the possibility to take and save images and video sequences [14, 15].

Compared to conventional US imaging, the quality of POCUS images can be worse. The affordability and portability of POCUS devices however still make them a great tool for settings where access to conventional US machines cannot be provided.

2.2 Breast Cancer

Breast cancer has been the most common type of cancer in recent years and surpassed lung cancer with an estimate of 2.3 million new cases in 2020, which makes up 11.7% of all cancer cases worldwide [1]. In females, it makes up 24.5% of all cancer cases and 15.5% of all cancer deaths, which is higher than any other cancer type. In 2020 an estimate of 875 thousand deaths were caused by breast cancer, with incidence, mortality and morbidity rates varying heavily between different countries and regions. While the incidence rates are significantly higher in transitioned countries compared to transitioning countries, the mortality rate is 17% higher in transitioning countries (15.0 per 100.000) as compared to transitioned countries (12.8 per 100.000), leading to disproportional survival rates [1]. Figure 1 shows region-specific age-standardized incidence and mortality rates for breast cancer.

The survival rates are particularly low for sub-Saharan African regions, which is largely due to bad health infrastructure [16, 17] and late-stage diagnosis [1]. The five-year age-standardized relative survival in 14 of those regions was 66.3% between 2008 and 2015. The relative survival after five year was the lowest in Kyadondo, Uganda, where only 12.1% survived [18], as opposed to survival rates of 85% or higher in many high-income countries [19].

The lack of access to breast cancer diagnostics is one of the main causes for low survival in low- and middle-income countries, therefore the aim of this study is to develop a safe algorithm

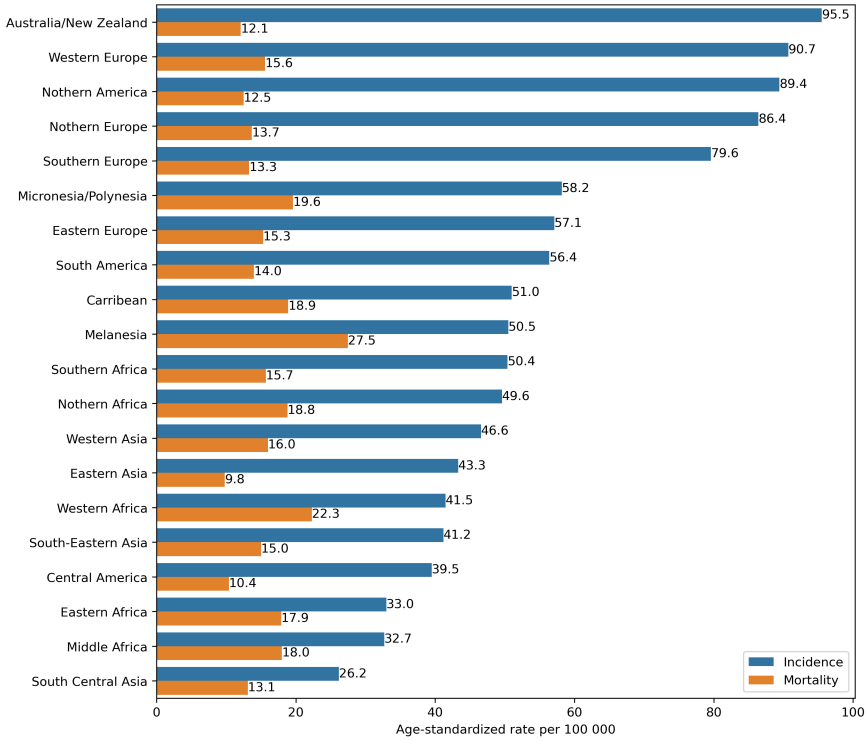

Figure 1 Age-standardized (world) region-specific incidence and mortality rates for breast cancer in 2020. The values were obtained from [1].

that can detect and classify breast cancer in POCUS images. This novel approach using low-cost, portable POCUS devices combined with a algorithm-driven diagnosis can be used to detect breast cancer early on, with the potential to significantly improve patient outcome.

2.2.1 Breast Cancer Characteristics

Breast cancer occurs when abnormal cells start growing in the breast, often leading to the formation of a tumor. A breast tumor can be malignant (cancerous) or benign (non-cancerous), where malignant tumor cells are spreading faster and can metastasize through the body. Benign findings include fibroadenomas, mostly found in younger women, and cysts. A typical symptom of breast cancer or benign lesions like cysts and fibroadenomas is having a lump, swelling or thickening of breast tissue. This, however, is not enough to classify which of the above it is and further diagnostic steps have to be taken.

In order to determine whether a tumour-like finding is malignant or benign, several characteristics found in US images can be evaluated. This includes lesion shape, orientation, margin, lesion boundary, echo texture, posterior acoustic feature, surrounding tissues, calcification and lesion vascularity [20]. A visual guide on how to determine the expression of some of the most important characteristics can be seen in Figure 2. Table 1 describes the appearance of the most important characteristics for cysts, fibroadenomas, cancer and gladular tissue in US imaging and Figure 3 shows corresponding US examples.

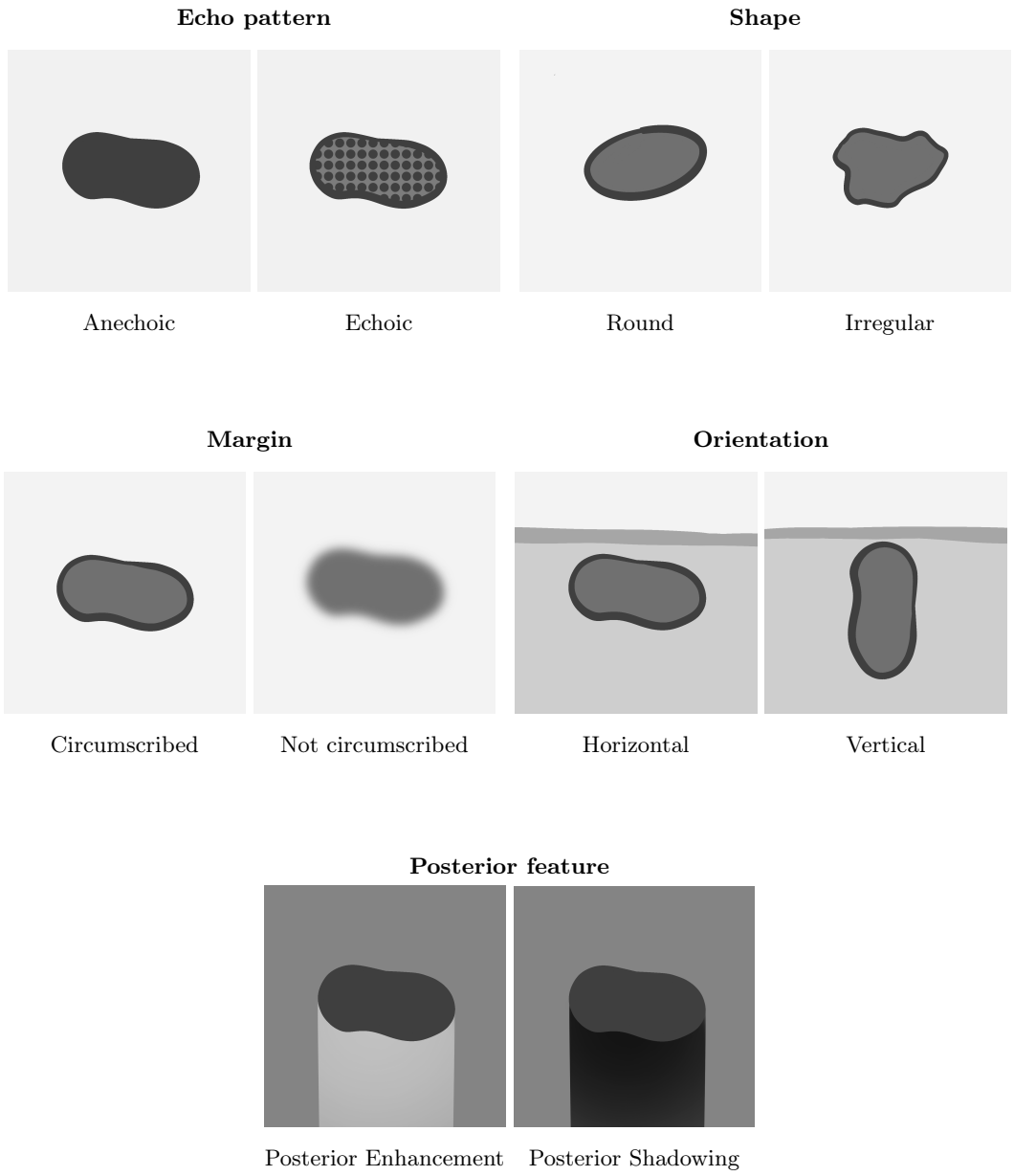

Figure 2 Breast lesion characteristics and expressions [21]. For each feature, the findings on the left are typically associated with benign lesions, whereas the expressions on the right are typically associated with malignant tumors.

Table 1 Characteristics for differentiation between malignant and benign findings in ultrasound imaging. The information was retrieved from [22]. Examples can be seen in Figure 3.

	Cyst	**Fibroadenoma**	**Cancer**	**Glandular tissue**
Echo pattern	anechoic pattern	hypoechoic, sometimes isoechoic	hypoechoic	hyperechoic
Shape	oval or round	most common: oval or round, less frequent: lobulated	most common: irregular shape, less frequent: round or oval	locally prominent glandular tissue
Margin	circumscribed	circumscribed, well-delineated	not circumscribed: indistinct, angular, microlobulated, spiculated	
Orientation	horizontal	horizontal (wider than tall, i.e. parallel to skin)	vertical (taller than wide, i.e. not parallel to skin)	
Posterior feature	posterior enhancement	sometimes (minimal) posterior enhancement	frequently posterior shadowing	no posterior feature
Calcifications	no calcifications	may have larger calcifications	may have small calcifications in or outside mass	no calcifications

The composition of breast tissue plays a heavy role in the appearance and quality of an US scan. The two types of breast tissue are glandular tissue and fatty tissue, with the ratio between them differing greatly. Influencing factors for the breast tissue composition include age, pregnancy and lactation, and adipositas. Due to the resulting US images looking very different, it can be hard to interpret the results and give a correct diagnosis. Examples of different breast tissue compositions and their effect on mammography and ultrasound imaging can be seen in Figure 4. This difference in images with varying tissue structures, patterns and large difference and contrasts, makes it especially hard for a DL algorithm to learn generalizable rules for inference.

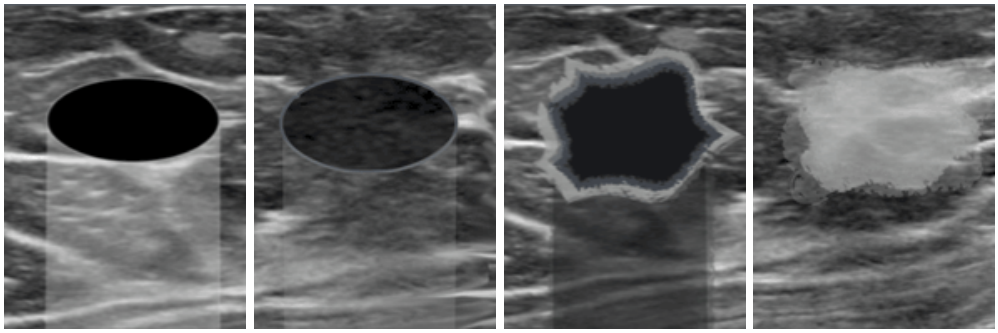

Figure 3 Exemplary typical characteristics in ultrasound imaging for different findings [22]. From left to right: Cyst, fibroadenoma, cancer, glandular tissue. The images are used with permission from *the Radiology Assistant*. Descriptions of the different characteristics are shown in Table 1.

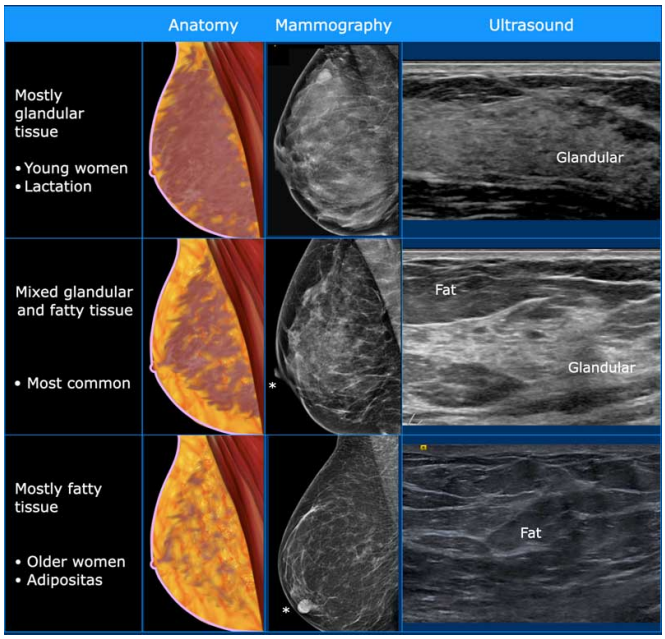

Figure 4 Breast composition and effect on mammography and ultrasound imaging [22]. The images is used with permission from *the Radiology Assistant*.

2.3 Deep Learning

Deep learning (DL) is a subfield of machine learning (ML), which is a subfield of artificial intelligence (AI), that consists of representation learning algorithms inspired by the human brain [23, 24]. The main objective is to learn how to process information and solve problems by learning from experience.

ML can be divided into supervised, unsupervised and reinforcement learning. In this context, learning refers to a model learning pattern from data. In supervised learning, training data exists together with true labels. Examples are classification tasks and regression tasks. In unsupervised learning, the data does not have labels, as it is the case in, for example, clustering. In reinforcement learning decisions are learned based on feedback and rewards, e.g. creating a smart agent for a game. DL can be used in all three learning settings. In the following, we will focus on supervised learning.

More specifically, DL is inspired by neurons and their interconnectivity, making it possible to learn complex functions and find underlying patterns in data.

2.3.1 Fundamentals of Deep Learning

The smallest building block of an artificial neural network (ANN) is an artificial neuron (AN), as seen in Figure 5. The AN can receive several inputs, all weighted by different weights, and a bias [25]. After summing these inputs up, they are given into an activation function, which will determine the output of the AN. This concept relates directly to neurons in the brain, which receive inputs through their dendrites and will fire (release an output) if the signals cross a certain threshold. While biological neurons only produce binary outputs, i.e. they fire or they do not fire, an AN can produce continuous outputs, depending on the activation function.

To form an ANN, several ANs are used and structured into layers [25]. An input layer is a layer that consists simply of inputs, while hidden layers and the output layer consist of ANs. Any layer is considered hidden if it is between the input and the output layer. A neural network is considered to be deep if it contains at least two hidden layers. The simplest form of

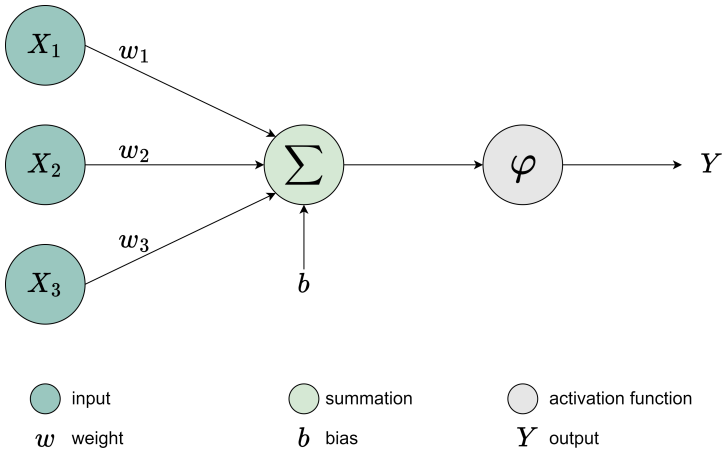

Figure 5 Artificial neuron with three inputs.

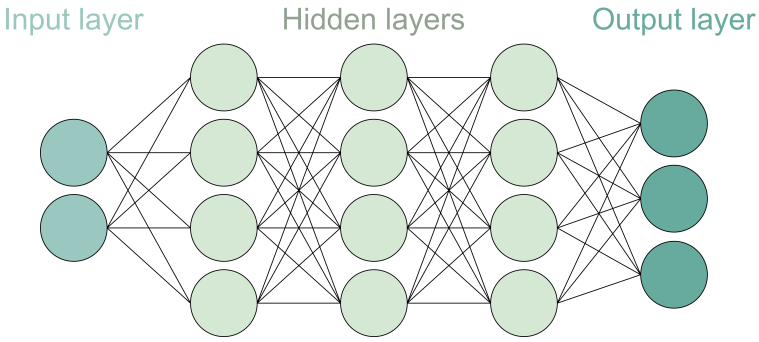

Figure 6 Deep neural network consisting of an input layer with two inputs, three hidden layers and an output layer with three outputs.

a deep neural network is a fully connected one, meaning that each AN in a layer is connected to each AN in the next layer (Figure 6). There are however many different types of ANNs, which can be suitable for different tasks and types of inputs. When working with images, typically a convolutional neural network (CNN) is used (see Section 2.4). DL is considered representation learning, as each layer in the network can be regarded a level of representation the data, with the input layer representing the raw data and the following layers representing more and more abstract versions of it, and the output layer representing the prediction for the respective problem [24].

2.3.2 Training Neural Networks

The performance of a neural network highly depends on finding suitable values for the weights and biases. Only with that, the input data can be processed in a way that underlying data structures are found and an accurate prediction can be formed. The purpose of training a neural network is therefore to find suitable weights and biases.

There are many parameters that need to be chosen for training an ANN. After setting the general network architecture, choosing the number of hidden layers, number of ANs in each layer, possible layer types (which might require more parameters) and activation functions, decisions about the training procedure and hyperparameters need to be made. This includes partitioning the available data into training, testing, and possibly also validation data. The network will only be trained on the training data, leaving other data to evaluate the performance independently on data that the network has not seen before. Typically, a validation data set is

used during hyperparameter tuning of the network, or for deciding which model to use as the final one from a pool of created models. A test set is only used as the final evaluation step, not influencing any training or decisions related to training and therefore being less biased. Validation sets are not always used, especially when only limited data is available, or when network architecture, parameters and training hyperparameters have already been decided and set as fixed.

When training an ANN, the goal is to find network configurations that yield accurate predictions for most inputs, both seen and unseen ones. When an input is passed feed-forward through a network, a prediction is calculated, which can be compared to the true label. In order to measure how good the prediction is, a *loss function* is used, which measures the distance between the predicted output and the ground truth. Using backpropagation, the loss is passed backwards through the network, allowing to update the weights and biases iteratively with the help of an *optimizer*. Optimizers are methods that define how the parameters will be updated with the aim to minimize the overall loss. The *learning rate* is a parameter that determines how much to change the weights and biases, i.e. how fast the network will learn from its error.

Further training hyperparameters include the *batch size* and *number of epochs*. The batch size determines how many input samples will be fed through the network before a backpropagation step updates the weights and biases. An epoch refers to a full cycle of using all batches of training data as inputs once. Therefore the number of epochs refers to how many times the full training data set is propagated through the network.

2.3.3 Evaluation Metrics

There are many different methods that can be used for evaluating the performance of an ANN. In the following, we will focus on evaluation methods for classification tasks. For a such a task, each data sample has a label, assigning it to be part of one class.

Confusion Matrix A confusion matrix represents the relation between true labels and predicted labels, and therefore gives a good overview of a model's performance. For a binary classification task, the predictions can be divided into true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Figure 7 visualizes a confusion matrix with TP, TN, FP and FN.

Based on the confusion matrix, several metrics can evaluate the performance, including *accuracy*, *precision*, *sensitivity* (also called *recall*) and *specificity*. The formulas are given based on the results shown in the confusion matrix:

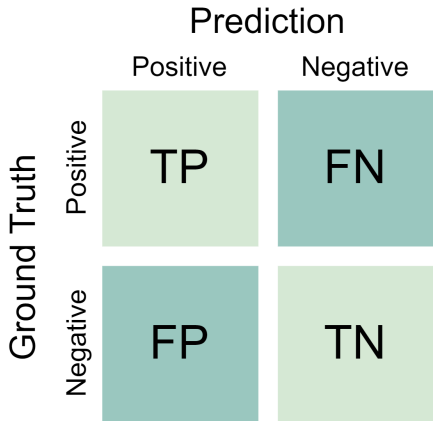

Figure 7 Confusion matrix displaying the relation between true labels and predicted labels.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

$$precision = \frac{TP}{TP + FP}, \tag{2}$$

$$sensitivity = \frac{TP}{TP + FN}, \tag{3}$$

$$specificity = \frac{TN}{TN + FP}. \tag{4}$$

An improved version of the accuracy is the *balanced accuracy*, which takes the number of samples per label into account and therefore is a more accurate measure on imbalanced data sets. The formula is given by

$$balanced\ accuracy = \frac{1}{N} \sum_{i=1}^{N} R_i, \tag{5}$$

with N being the number of classes and R_i being the sensitivity (recall) of class i. In multi-class classification problems, the confusion matrix is larger, but the idea between the metrics stays the same, with the accuracy being the number of correct predictions divided by the total number of predictions. In the scope of this thesis the balanced accuracy will be used as a metric. For simplicity, we will simply refer to this as accuracy.

ROC Curve The *Receiver Operating Characteristic* (ROC) curve displays the performance of a binary classifier for different classification thresholds. More specifically, it plots the sensitivity (true positive rate) in dependence of 1 − specificity (false positive rate). An example of ROC curves for different classifiers is shown in Figure 8.

AUC The area under the curve (AUC) is a measure of the area under the ROC curve and gives inside into the performance of a network. A random classifier has an AUC value of 0.5, and a perfect classifier would have an AUC of 1.

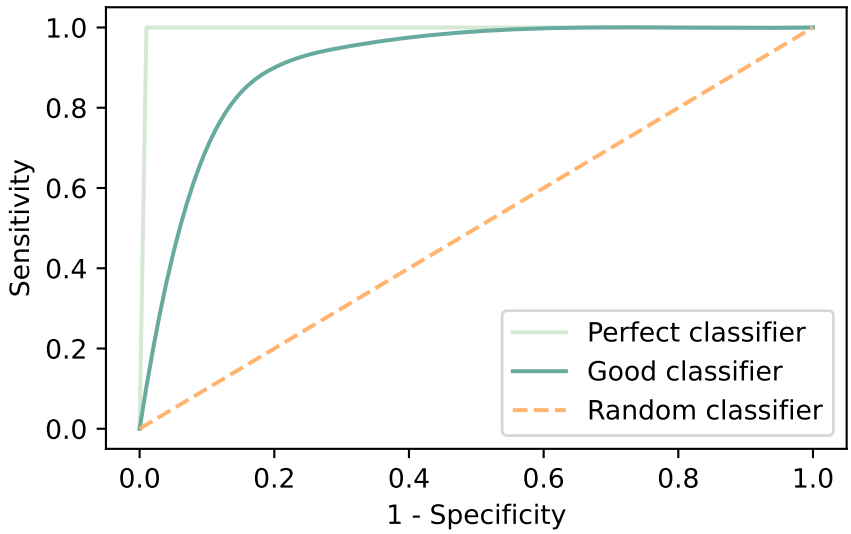

Figure 8 ROC curves of a perfect, good and random classifier.

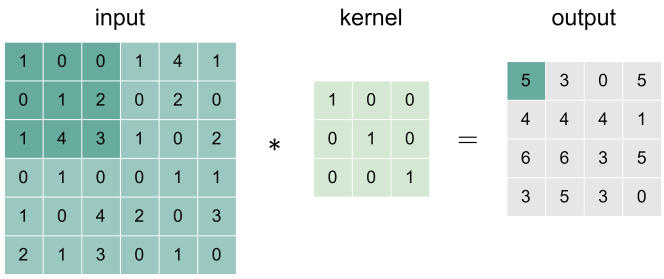

Figure 9 Example of convolution. For simplicity, the kernel has already been doubled flipped. For each patch of the input, element-wise multiplication with the double flipped kernel is performed. The sum of it determines the output for that patch.

2.4 Convolutional Neural Networks

A convolutional neural network (CNN) [26] is a type of ANN working on grid-like data structures, most commonly used for processing images. Instead of having only fully connected feed-forward layers, a CNN also has convolutional layers. Processing data in these layers allows for taking spatial information into account. This is done via the processing of small patches (grids) together.

The base of CNNs is the mathematical concept of convolution. A matrix is processed with the help of kernel, forming another matrix. Given an image H and a kernel K of size $M \times N$, the result of the convolutional operation at location (x, y) in the output matrix is given by

$$(H * K)(x, y) = \sum_{i=-\frac{M}{2}}^{\frac{M}{2}} \sum_{j=-\frac{N}{2}}^{\frac{N}{2}} H(x - i, y - j)K(i, j). \tag{6}$$

In other words, element-wise multiplication between the double flipped kernel and the patch of the image of the same size as a kernel, centered around the given location, is performed and summed up. A step size (stride) needs to be set that determines how much the kernel moves after every step. Typically, this is one, meaning that the kernel will not skip anything. Furthermore, padding can be added around the border of the input in order for the kernel to be able to reach pixels furthest out. A visual example of convolution on a 6×6 matrix with a 3×3 kernel with step size 1 and no padding is shown in Figure 9. In a convolutional layer of a CNN, the kernels replace typical ANs, producing feature maps instead of single-valued outputs. The kernel values are the weights that will be learned during training. A convolutional layer usually has several kernels.

The second important layer type in CNNs are pooling layers. As opposed to convolutional layers, which create feature maps, pooling layers make each of the feature maps smaller. This is achieved by, similar as in convolution, performing an operation on patches of the input like averaging or taking the maximum value. Pooling layers are typically used after convolutional layers, often with size 2×2 and strides 2, reducing each feature map size drastically. An example of a CNN with two convolutional layers and two pooling layers is shown in Figure 10. After the last layer working with feature maps, the data needs to be flattened before feeding it into a fully connected network, consisting of at least one fully connected layer, which will perform the final classification based on the previously extracted features.

Other important layers include dropout layers, which randomly turn off a certain amount of ANs each time during training. This means that the corresponding connections will not be used, which ensures that the network will not focus too strong on only a few connections, but rather many connections will impact the final prediction.

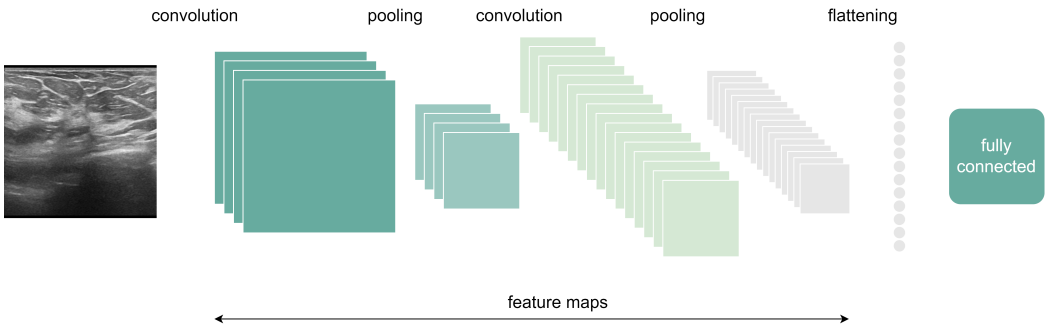

Figure 10 Convolutional neural network consisting out of two convolutional blocks with each a convolutional layer and a pooling layer, followed by a flattening layer and finally a fully connected network for classification.

2.5 Uncertainty

In recent years there has been an increasing interest in identifying and quantifying the predictive uncertainty of neural networks [27]. As an ideal solution, a deep learning model should not only predict a class label during a classification task, but also output how certain it is about the respective classification. In a regression task, it additionally should output confidence intervals. This new feature of being able to quantify the uncertainty becomes especially important in applications where a wrong classification will potentially lead to severe consequences, e.g. life-threatening outcomes in medical applications [27].

Uncertainty in DL models can be divided into two categories: aleatoric and epistemic uncertainty [8]. While aleatoric uncertainty refers to the uncertainty coming from the data itself, epistemic uncertainty is the uncertainty that stems from the model. The predictive uncertainty of a classification produced by a neural network is defined as the sum of of aleatoric and epistemic uncertainty:

$$\mathcal{U}_{prediction} = \mathcal{U}_{aleatoric} + \mathcal{U}_{epistemic}. \tag{7}$$

2.5.1 Aleatoric Uncertainty

Aleatoric uncertainty in a DL model is the uncertainty that stems from the noise or variability in the data and is therefore irreducible by nature [28]. Independent of how many samples will be collected for a data set, the aleatoric uncertainty of the prediction cannot be reduced. This inherent randomness can e.g. origin in sensor noise during measuring. Another possible source is the natural variability of the model's input, which is the variability of the data and its properties itself (including material properties and compositions that can slightly differ, environmental conditions etc.). The aleatoric uncertainty will also increase when it becomes unattainable to separate between different classes (e.g. due to dimensionality reduction or natural non-separability). In this case the input data already presents a lack of predictive power [27].

Aleatoric uncertainty can be divided into two different types of data uncertainty: Homoscedastic uncertainty and heteroscedastic uncertainty. Homoscedastic uncertainty is constant for all samples in the data set, whereas heteroscedastic uncertainty can differ between samples [29].

2.5.2 Epistemic Uncertainty

Epistemic uncertainty is the model uncertainty or approximation uncertainty which is caused by the model's lack of knowledge about the true underlying data distribution, the class boundaries

and the correct model parameters for a perfect predictor. The epistemic uncertainty is therefore reducible by nature [28] and will decrease given more data.

There are two major sources that create epistemic uncertainty in a prediction: Model-form uncertainty and parameter uncertainty [27]. Epistemic model-form uncertainty is caused by the structure of the chosen model, including model architecture type, layers and activation functions. These might not be able to fully capture underlying structures for classification. Epistemic parameter uncertainty is caused by the training process of the network, which might lead to non-optimal model parameters or only local optima. These non-optimal parameters can occur due to limited training data, biased or imbalanced training data or low training data fidelity. This makes it harder or impossible for the training algorithm to find parameters that create an accurate predictor.

Measuring the epistemic uncertainty can be especially useful for detecting OOD samples. For ID samples, a UQ calculation method should have small epistemic uncertainties, which should be clearly differentiable from OOD samples with high epistemic uncertainties [6]. One reason for this is the low covariance between OOD data and training samples [27].

2.5.3 Mathematical Foundations of Uncertainty Decomposition

A Bayesian approach can be used to define the general formulas that hold for uncertainty [30, 31]. Given a model with parameters θ, where θ is a realization of the stochastic variable Θ, and unobserved data D, we can use Bayes' theorem

$$\mathbb{P}(\Theta|D) = \frac{\mathbb{P}(D|\Theta)\mathbb{P}(\Theta)}{\mathbb{P}(D)}, \tag{8}$$

where $\mathbb{P}(\Theta)$ is the *prior distribution* (the prior belief we have about the model parameters), $\mathbb{P}(\Theta|D)$ is the *posterior distribution*, $\mathbb{P}(D|\Theta)$ is the *likelihood* and $\mathbb{P}(D)$ is the *marginal*, or also called the *evidence*. After obtaining the posterior or an approximation of it, the predictive distribution for a class y given test input sample x and training data D can be derived as

$$\mathbb{P}(y|x, D) = \int \mathbb{P}(y|x, \Theta = \theta)\mathbb{P}(\Theta = \theta|D)d\theta. \tag{9}$$

The aleatoric uncertainty is the uncertainty from $\mathbb{P}(y|x, \Theta = \theta)$, and the epistemic uncertainty is the uncertainty from $\mathbb{P}(\Theta = \theta|D)$.

Different uncertainty metrics can be used to perform further calculations. Given an uncertainty metric $\mathcal{I}$, the total uncertainty, aleatoric uncertainty and epistemic uncertainty can be calculated as follows

$$\mathcal{U}_{total} = \mathcal{I}[\mathbb{E}_{\theta\sim\mathbb{P}(\theta|D)}[\mathbb{P}(y|x, \theta)]] = \mathcal{I}[\mathbb{P}(y|x, D)], \tag{10}$$

$$\mathcal{U}_{aleatoric} = \mathbb{E}_{\theta\sim\mathbb{P}(\theta|D)}[\mathcal{I}[\mathbb{P}(y|x, \theta)]], \tag{11}$$

$$\mathcal{U}_{epistemic} = \mathcal{U}_{total} - \mathcal{U}_{aleatoric}. \tag{12}$$

While for regression tasks variance is a typically uncertainty metric to use [32], for classification tasks, entropy is commonly used [31]. For the distribution of a label y for data sample x, the entropy is defined as

$$\mathcal{H}[\mathbb{P}(y|x, D)] = -\sum_{c=1}^{K} \mathbb{P}(w_c|x, D)\log\mathbb{P}(w_c|x, D), \tag{13}$$

where w_c is the probability of sample x belonging to class c.

2.6 Uncertainty Quantification

With the increased employment of AI-based algorithms for decision making in a vast variety of fields, the importance of trust and safety becomes a crucial theme [7, 33]. This becomes especially important in fields like medical diagnostics or self-driving cars, where errors can lead to severe consequences and life-threatening outcomes. As a result, the field of UQ emerged, which has gained growing attention since approximately 2015 [27]. In recent years, many UQ methods have been proposed and applied to various fields, including medical image analysis, image processing, computer vision, bioinformatics and natural language processing [34]. These methods aim at detecting precisely those uncertainties.

Access to quantified uncertainties can help with determining whether a prediction can be trusted and when to be extra cautious. For an optimal UQ method high uncertainties indicate that the model is not sure about its prediction. A combined interpretation of a model's prediction together with UQ results is therefore considered to strengthen its transparency and trustworthiness [27]. Additionally, access to reliable uncertainties shows great potential for optimization purposes [34]. This means that training settings and network architectures can be changed accordingly, as well as data cleansing and pre-processing, if the uncertainties show specific patterns for specific inputs.

In order to build a trustworthy classifier, it is therefore crucial to look at the uncertainties within the predictions. There are different methods for quantifying these uncertainties, some of which require whole new network structures, and others which can be used post training of a classical, frequentist model. Two widely used UQ approaches include probabilistic AI using Bayesian approximation and neural network ensembles. While Eq. 7 generally holds, not all uncertainty quantification methods are able to differentiate between aleatoric and epistemic uncertainty.

2.6.1 Out-of-Distribution Detection

One subtask that lies within the field of UQ is the detection of out-of-distribution (OOD) data. OOD samples are data samples that the model has not learned how to process or interpret reliably, as they are too different to the training data. A trustworthy model should therefore possess the ability to detect such samples and flag or discard them [7, 35].

In OOD detection, the training data is considered to be in-distribution (ID). Typically, it is assumed that test data will be part of the same distribution. In practice, this is however not always the case and test samples can in fact be OOD due to various reasons. OOD detection is a field which aims at developing methods that are able to detect such samples.

3

Data

In the scope of this work several data sets have been used. The data sets can be divided in ID data sets and OOD data sets. The ID data sets have been used for training all networks and for evaluation of the UQ experiments. The OOD data sets have been used for evaluation of the OOD detection experiments.

3.1 In-Distribution Data

The ID data consists of POCUS and conventional US images capturing breast tissue. The POCUS data was collected using a GE Vscan air probe [36]. The US images were collected using the conventional ultrasound machines Logiq E9 and Logiq E10 [37]. All images have been recorded at Skåne University Hospital, Malmö, and labeled by radiologists. The data is classified into malignant lesions, benign lesions and normal breast tissue. Examples of POCUS images for each class can be seen in Figure 11, examples of US images for each class are displayed in Figure 12.

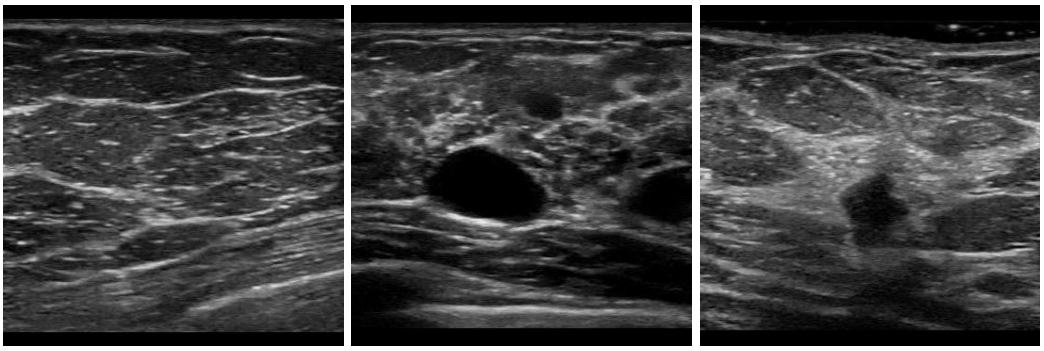

Figure 11 POCUS images capturing normal tissue, benign and malignant lesions (from the left to right).

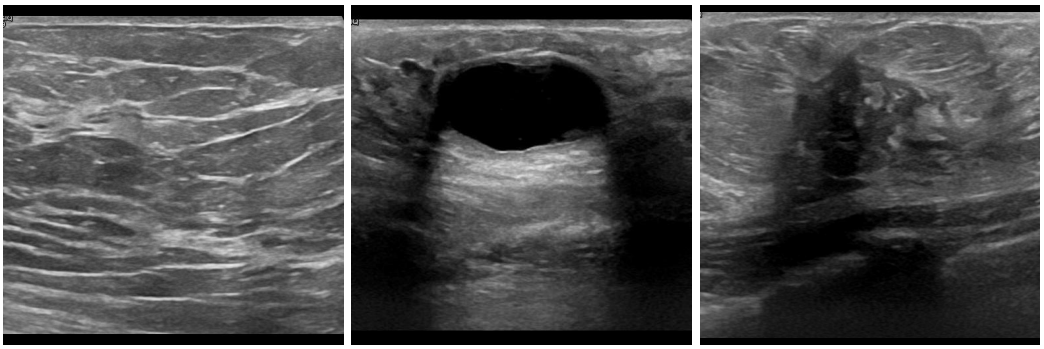

Figure 12 US images capturing normal tissue, benign and malignant lesions (from left to right).

Table 2 Sizes of the ID data sets.

		POCUS	US	Total	Data set size
Train	Normal	463	386	849	1974
	Benign	173	254	427	
	Malignant	178	520	698	
Test	Normal	284	-	284	531
	Benign	131	-	131	
	Malignant	116	-	116	
Total		1345	1160	2505	

The training data set consists of both POCUS and US images, while the test data set only contains POCUS images. This was decided since the inference only has to work with POCUS data later. The US data was therefore added to the training set due to limited (training) data. Table 2 shows the sizes of different data sets. Both the training and the testing data sets are imbalanced, with fewest training examples existing for the class benign and by far most testing images existing for the class normal.

3.2 Out-of-Distribution Data

For the OOD detection experiments, three different OOD data sets have been used: MNIST, CorruptPOCUS and CCA. All three data sets contain images that are different to the images from the ID training data set.

MNIST The first OOD data set is MNIST, which was first introduced by LeCun et al. [38] in 1998. Here, we only use the MNIST test set, which contains 10 000 grayscale images of handwritten digits. For simplicity, we will refer to this OOD test set just as MNIST. It does not resemble the ID data much and is therefore used as a baseline. Examples of images from the MNIST OOD data set are displayed in Figure 13.

CorruptPOCUS The second OOD data set used in the scope of this work is CorruptPOCUS. This data set is based on the ID test data set of POCUS images described in Section 3.1 and made to resemble POCUS OOD data that can occur naturally. CorruptPOCUS was created by Karlsson et al. [39] and consists of distored versions of the 531 images from the POCUS test set. The images were modified by adding dark areas, noise and blur. Figure 14 shows examples of images from the CorruptPOCUS OOD data set.

CCA The last OOD data set used in this work is CCA. This data set consists of 84 conventional ultrasound images capturing the common carotid artery [40]. The images were captured with a Sonix OP ultrasound scanner and are OOD due to not displaying breast tissue. This data set most realistically resembles POCUS images of class normal. Images from the CCA OOD data set can be seen in Figure 15.

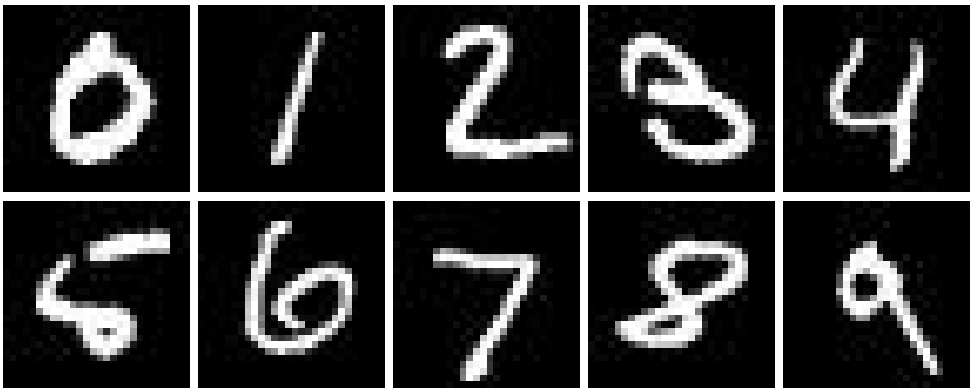

Figure 13 Example images from the MNIST data set.

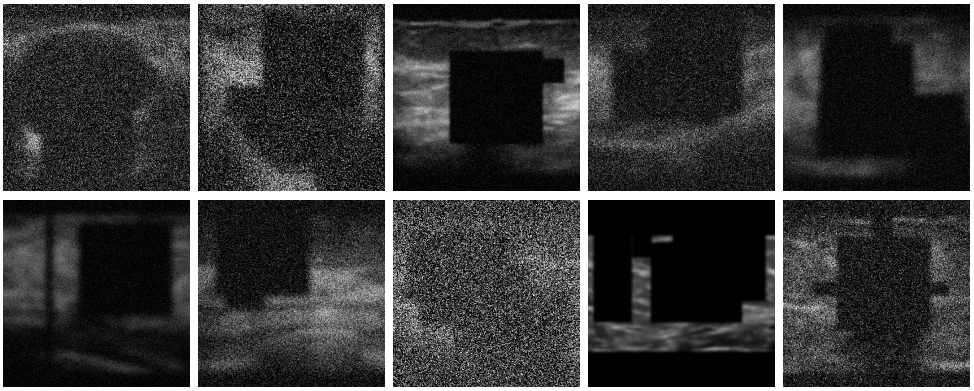

Figure 14 Example images from the corruptPOCUS data set. The images have been generated by distorting the POCUS images from the ID test set.

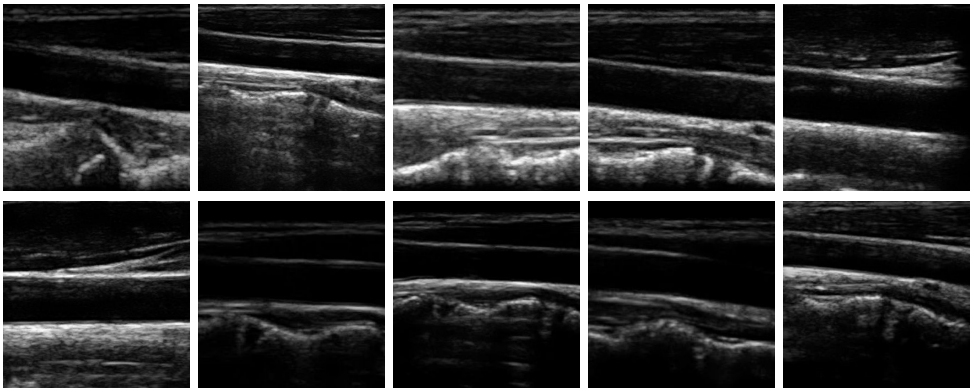

Figure 15 Example images from the CCA data set. The images are capturing the common carotid artery and have been recorded with conventional US.

4

Theory of Methods

In the following chapter, the theory of the different UQ and OOD detection methods used in this work will be explained. This includes BNNs, neural network ensembles and the post-hoc methods softmax output, trust score and energy score. Detailed descriptions on how to calculate uncertainties using each method will be given.

4.1 Bayesian Neural Networks

As opposed to a non-Bayesian (frequentist) neural network, which has fixed values as weights, a BNN's weights are represented by a probability distributions (Figure 16). These probability distributions contain information about the probabilities of a range of possible values [41]. Moreover, BNNs are robust to overfitting problems [34] and have shown to produce similar performance results to frequentist neural networks [29].

4.1.1 Fundamentals of Bayesian Learning

Since there are no fixed point estimates for the weights that can produce a prediction $P(\hat{y}|x)$ in a Bayesian network, the prediction for class $\hat{y}$ for sample x is formed as the mean over M predictions with different weights w, using a Monte Carlo sampling approach:

$$P(\hat{y}|x) = \frac{1}{M} \sum_{w} P(\hat{y}|w, x), \tag{14}$$

where the weights w and sampled from the posterior distribution $\mathbb{P}(w|D)$, with D being the training data. The main purpose in Bayesian learning therefore is to find the posterior distribution of the weights that fits the data. The foundation of Bayesian learning is the Bayes'

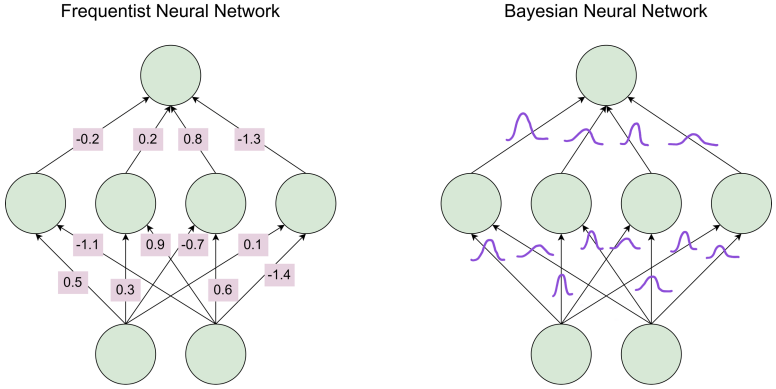

Figure 16 Difference between weights in deterministic and probabilistic neural networks. The network on the left has deterministic fixed-valued weights, like a normal CNN. The weights of the network on the right are represented by probability distributions, like a BNN.

Table 3 Comparison of different methods for solving the posterior in Baysian Neural Networks for Uncertainty Quantification [34, 27].

	Advantages	Disadvantages
MCMC	• nonparametric • high accuracy • asymptotically correct	• slow • not suitable for large data sets • not suitable for complex models
VI	• fast • suitable for large data sets • maximizes an explicit objective	• hard to train (depends on starting condition) • complex to calculate
MCD	• very easy to implement • easy to train	• not good at OOD detection

theorem (Eq. 8), which can be used to define how to obtain the posterior distribution of the weights. Using a Bayesian approach, prior knowledge and beliefs are directly incorporated into the prediction. Applying Bayes' Theorem to BNNs leads the following equation:

$$\mathbb{P}(w|D) = \frac{\mathbb{P}(D|w)\mathbb{P}(w)}{\mathbb{P}(D)}. \tag{15}$$

In Bayesian inference, the aim is to calculate the posterior distribution of the weights given the training data, denoted as $\mathbb{P}(w|D)$. Given that, the distribution of a possible label $\hat{y}$ for a data sample x is given by

$$\mathbb{P}(\hat{y}|x) = \mathbb{E}_{\mathbb{P}(w|D)}[\mathbb{P}(\hat{y}|w,x)]. \tag{16}$$

To calculate this expected value, all possible combinations of weights need to make a prediction on $\hat{x}$, which would be weighted according to the posterior distribution. As this implies an infinite number of network configurations, inference and learning in BNNs is intractable. To overcome this, approximate inference needs to be performed.

While in theory any types of distributions can be used in approximate inference, Gaussian distributions are often used in practice. They come with the advantage of closedness properties and being easy to learn, only having two parameters, i.e. mean and variance. Compared to frequentist neural networks, where there is only one value per weight, a Bayesian network with Gaussians therefore doubles the amount of parameters.

There are different approximation methods for solving the Bayesian posterior, which include Markov chain Monte Carlo (MCMC), variational inference (VI) and Monte Carlo dropout (MCD). The different methods are desribed more in detail in following sections, and a comparison of advantages and disadvantages of each method for the purpose of UQ is shown in Table 3.

4.1.2 Markov Chain Monte Carlo

To solve the Bayesian posterior, MCMC uses posterior sampling. MCMC creates a Markov chain which converges to the true posterior distribution of the weights. The most commonly used MCMC algorithms for BNNs are the Metropolis-Hastings algorithm [42, 43] and Hamiltonian Monte Carlo [44].

While in theory MCMC methods are very appealing due to their convergence to the true posterior, their execution fails for larger data sets and higher dimensions [45]. This leads to them only being used little in BNNs.

4.1.3 Variational Inference

Opposed to MCMC, VI uses variational approximation to solve the Bayesian posterior. This approach has previously been studied by Hinton et al. [46], Graves [47] and Blundell et al. [41] and is the most commonly used approach to implement BNNs. Specifically, many modern implementations of VI are based on *Bayes by backprop* [41].

The principal objective of VI is to approximate the posterior distribution of the weights and iteratively improving it so that the approximation is as close as possible to the true posterior. Intuitively, the parameters of the weight distributions need to be found that match the underlying true distribution the best. This is achieved by minimizing the distance between the approximation and the true posterior using the Kullback-Leibler (KL) divergence. The KL divergence is defined as

$$\mathrm{KL}[P||Q] = \int P(x) \log\left(\frac{P(x)}{Q(x)}\right) dx, \tag{17}$$

and measures how similar the two distributions are, with a low KL divergence value representing that the distributions are similar. In order to find parameters θ^* for a good approximation of the posterior distribution, the KL divergence therefore needs to be minimized

$$\theta^* = \arg\min_\theta \mathrm{KL}[q(w|\theta)||\mathbb{P}(w|D)], \tag{18}$$

where $q(w|\theta)$ is the approximation of the true posterior distribution $\mathbb{P}(w|D)$ of the weights w with the distribution parameters θ. When using Gaussian distributions, θ consists of a mean and a variance value. In order for the BNN to learn based on KL divergence of the approximation and the true posterior distribution, Eq. 18 can be reformulated into a loss function

$$\mathcal{L}(\theta, D) = \mathrm{KL}[q(w|\theta)||\mathbb{P}(w)] - \mathbb{E}_{q(w|\theta)}[\log \mathbb{P}(D|w)], \tag{19}$$

which is called the expected lower bound (ELBO) loss, or in other sources also the variational free energy. The Bayes by backprop method [41] futhermore suggests to approximate the loss using

$$\mathcal{L}(\theta, D) \approx \sum_{i=1}^{n} \log q(w^{(i)}|\theta) - \log \mathbb{P}(w^{(i)}) - \log \mathbb{P}(D|w^{(i)}), \tag{20}$$

with $w^{(i)}$ being the i-th weight sample drawn from the variational posterior distribution of the weights. This simplified cost function can be used together with gradient descent to update the parameters of the weights iteratively during training of the BNN. The training is hence similiar of the one of a frequentist neural network using backpropagation, except that instead of point estimates, there are several parameters that need to be updated for each weight.

4.1.4 Monte Carlo Dropout

MCD [48] uses random dropout of neurons of the neural network. This can be achieved by using dropout layers in a neural network that are active both during training and inference. Dropout is generally considered a regularization method for neural network training that can prevent overfitting, but it can also be used in the context of Bayesian learning. Due to dropout always choosing random neurons to not use, the outputs will differ every time a prediction is

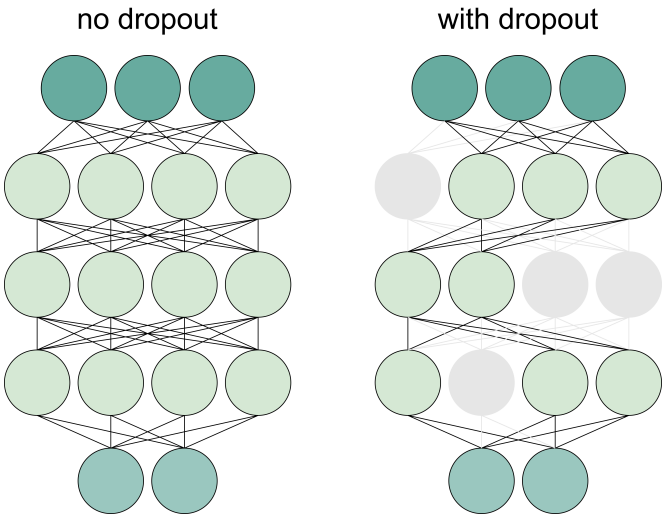

Figure 17 Neural network with and without dropout. If dropout is used, a random certain amount of neurons is deactivated each time. How many neurons will be deactivated is defined for each layer separately.

made. Figure 17 shows a visualisation of the dropout technique. The more times a prediction for the sample will be made, the more the outputs will converge to a certain distribution. Due to that, MCD is considered a Bayesian method, however it does not use distributions for each weight, nor allow to include prior knowledge, and rather shows a different behavior than other Bayesian methods [49, 50]. It is therefore highly debated if it should be classified as a method for BNNs or as something different instead, e.g. an ensemble method.

4.1.5 Uncertainty in Bayesian Neural Networks

Uncertainty in BNNs can be measured by sampling several predictions with the weights from the posterior distribution. It can then be decomposed further into aleatoric and epistemic uncertainties.

The aleatoric and epistemic uncertainty in a prediction for a data sample x^*, using a BNN with weights w, can be calculated by sampling predictions N times:

$$\mathcal{U}_{aleatoric} = \frac{1}{N}\sum_{i=1}^{N} \text{diag}(\hat{p}_i) - \hat{p}_i\hat{p}_i^T, \tag{21}$$

$$\mathcal{U}_{epistemic} = \frac{1}{N}\sum_{i=1}^{N} (\hat{p}_i - \bar{p})(\hat{p}_i - \bar{p})^T, \tag{22}$$

where $\bar{p} = \frac{1}{N}\sum_{i=1}^{N}\hat{p}_i$ is the mean prediction and $\hat{p}_i = \text{Softmax}(f_{w_i}(x^*))$ is the prediction for sample x^* using the BNN f with weights w_i.

4.2 Neural Network Ensembles

The idea behind neural network ensembles, or deep ensembles, is to independently train multiple models on the same problem and then use all their predictions to make a final, more informed prediction. They have first been introduced in 1990 by Hansen et al. [51] and have since then found great use in a variety of applications. Neural network ensembles can prevent overfitting [27] and improve the generalizability of a neural network [52]. They have previously been used to study predictive uncertainties [6] and perform OOD detection [53, 54, 55]. The general design of a neural network ensemble can be seen in Figure 18.

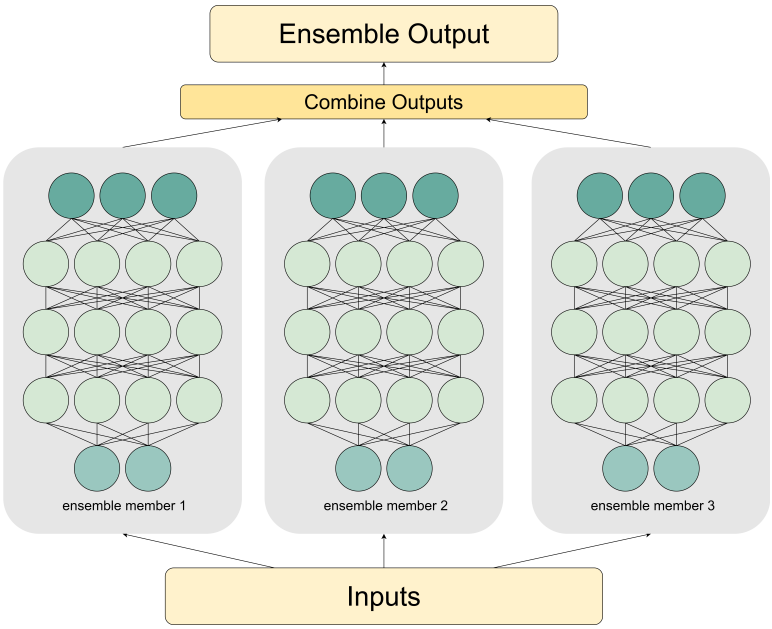

Figure 18 Design of a neural network ensemble. An ensemble consists of a finite number of ensemble members, with each of them being a neural network. All networks get the same input, each producing their own output. A method for combining the individual outputs is applied to form the final output of the ensemble.

4.2.1 Deep Ensembling Techniques

There are different methods on how to combine different models into one ensemble. The models that form the ensemble are called ensemble members. Some of the most common methods are an average ensemble, a weighted average ensemble, a plurality vote ensemble and a concatenation ensemble (Figure 19). While averaging is typically used in regression task, plurality vote is often used in classification tasks.

Average Ensemble The easiest and most intuitive approach to ensembling different neural networks trained on the same problem is to average their outputs into one single prediction. While this is a good approach for regression tasks, it can be less effective for classification tasks. In such, there are outputs for each of the possible classes. An average ensemble then calculates the average output for each class, which will possibly introduce more uncertainty and lead to a loss of information, since high and low values will cancel each other out. Another disadvantage is that each ensemble member will contribute to an equal extent, independent of the accuracy of each member. If the ensemble consists of very diverse networks, which might be specifically trained on sub-problems or perform especially well on a subset of classes, the averaging process will lose crucial information.

Weighted Average Ensemble The weighted average ensemble improves the concept of a simple average ensemble by adding weights to the different ensemble members. These weights can either be per ensemble member or per each output of each member.

There are different ways on how to determine and optimize the weights of a weighted average ensemble. One simple approach is to weigh each ensemble by its standard deviation on the test data. This can be done for each class separately or combined. The final classification will then be the average of each model's output, weighted by the model-specific weights.

Plurality Vote Ensemble A plurality vote ensemble does not work with the raw softmax outputs of the ensemble members, but rather with the predicted classes from each ensemble

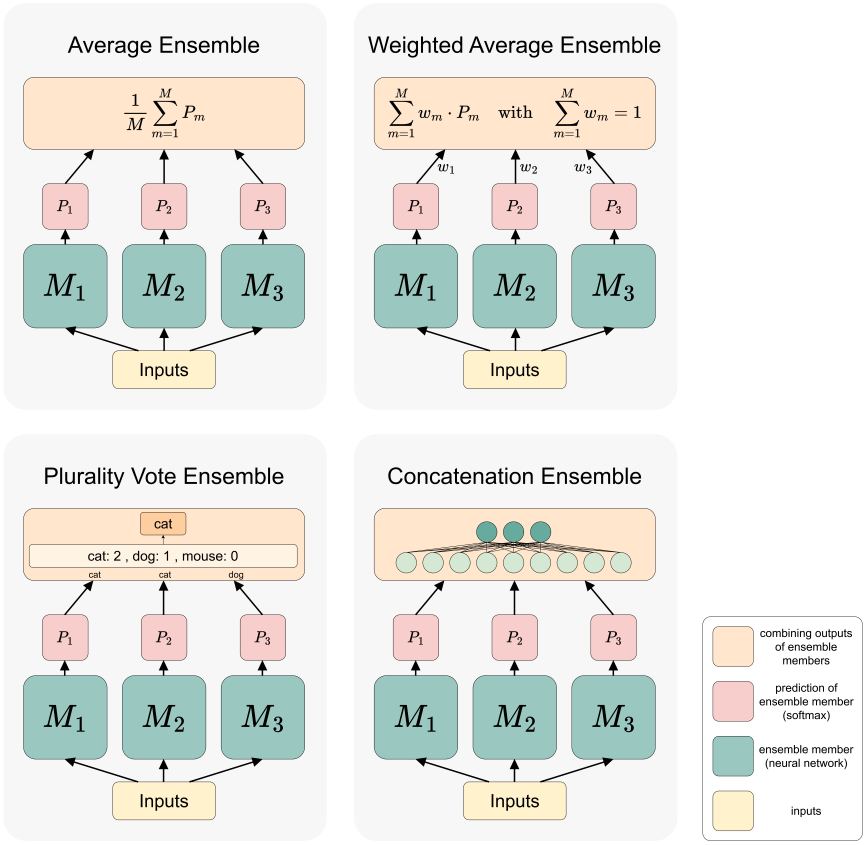

Figure 19 Different ensemble types. Each type differs in the way the outputs of the ensemble members are combined in order to make the final prediction.

member. To determine the final class prediction, the class with the highest number of votes from the ensemble members is picked. This overcomes some of the problems present in average ensembles, but it does not account for individual member's accuracies. In the case of a tie between two classes, it is unclear how the ensemble should choose a label and usually a random pick is made.

Concatenation Ensemble In a concatenation ensemble, outputs from the ensemble members are flattened and concatenated into one large vector. This can either be the softmax outputs, the logits (raw values before the activation function) from previous parts of the network, or a mixture of both. As the inputs are stacked, dimensions do not matter, making this a more diverse method. A common approach is to simply use the logits from the dense layer before the softmax activation function, or the softmax probabilities. After the concatenation, a small classification network can be added. Typically, this consists of a single fully connected layer with the output corresponding to the classes, and needs to be trained after the ensemble members have finished training.

While this is a common deep ensembling technique for classification problems, it is not always the most suitable. One major disadvantage compared to an average or weighted average ensemble is the computational cost. Depending on the number of ensemble members and the input sizes to the concatenation layer, the dimension can become impractically high and a different ensembling technique may be more suitable. However, a concatenation ensemble has the advantage of being able to learn the weights of the inputs from the different ensemble members in a more sophisticated way than weighted averaging can do. For a classification task, it is therefore the most promising approach.

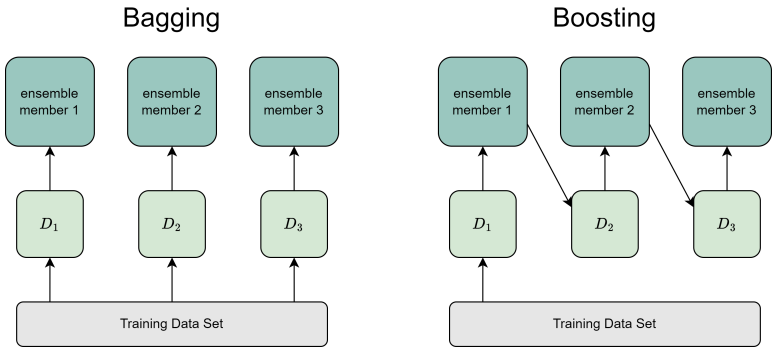

Figure 20 Bagging vs. boosting for training neural network ensembles. Bagging is a parallel process, where each ensemble members gets a random subset of the training data set to train on. Boosting is a sequential process, where the training data set for each ensemble member is influenced by the performance of the previous ensemble member.

4.2.2 Training Ensembles

The aim of the different ensemble members is to have expertise in different areas of the respective problem, so that combining their knowledge will lead to better results than just having one network. This is based on the fact that there is no optimal set of weights for a neural network that yields a perfect performance, but rather there are a lot of locally optimal solutions [51]. These weights that find the locally optimal solutions can highly depend on random factors like weight initialization and order of training samples, but are also affected by the parameter choices for e.g. batch size, learning rate, optimizer etc. It is therefore important to diversify the separate models in an ensemble during training [56] to ensure that each model finds a different solution to the problem, making errors for different subparts of the data set, while still all performing well. This diversification can be achieved via several different training setups and methods. The two most popular methods for training ensembles are bagging [57] and boosting [58] (Figure 20).

Bagging is short for bootstrap aggregating and is based on the idea that the ensemble members should be trained independently or in parallel. During training, each ensemble member is trained on a random subset of the available data, ensuring that the each model learns differently and the overall confidence and correctness of the prediction should increase. The random subsets are called bags and can be designed in different ways. A simple approach is to randomly pick a subset of data for each ensemble member, with or without replacement. At this step, augmentation can also be used to randomize the subsets more.

Opposed to bagging, boosting is based on the idea that the ensemble members are trained sequentially, with the previous results impacting the training of the next ensemble member. Essentially, the next training data set is influenced by the performance of the previous model, making it more important to pick data samples that have been misclassified before. This should ensure that the bias will partly be reduced and that the next model will focus more on a subpart of the original problem that has not been solved properly yet.

Apart from introducing randomness in the training set, diversification can also be achieved by using different training setups for each model. This can include using different learning rates, optimizers, numbers of epochs, batch sizes, or even the number of layers and general network architecture. It is however crucial to pick these only from a reasonable range so that resulting classifiers still perform well.

4.2.3 Uncertainty in Deep Ensembles

The uncertainty in an ensemble can be measured using Eq. 10. The uncertainty calculations are based on the outputs from the different ensemble members and are therefore calculated the same

way independent of the ensembling technique. Since we are performing a classification task, the uncertainty metric used is entropy (Eq. 13), and the total predictive uncertainty becomes

$$\mathcal{U}_{total} = -\sum_{c=1}^{K} F_c(x|\theta)\log F_c(x|\theta), \tag{23}$$

with $F_c(x|\theta)$ being the prediction for class c of ensemble F, with this prediction being formed by the different ensemble members according to the following

$$F(x|\theta) = \frac{1}{M}\sum_{m=1}^{M} f_m(x|\theta_m). \tag{24}$$

The prediction F is based on the separate predictions of M models that are weighted equally. Here, f_m is the m-th ensemble member and θ_m are the corresponding model parameters. Consequentially, the aleatoric and epistemic uncertainties can be derived using Eq. 11 and Eq. 12.

$$\mathcal{U}_{aleatoric} = -\frac{1}{M}\sum_{m=1}^{M}\sum_{c=1}^{K} f_{m_c}(x|\theta_m)\log f_{m_c}(x|\theta_m), \tag{25}$$

$$\mathcal{U}_{epistemic} = -\sum_{c=1}^{K} F_c(x|\theta)\log F_c(x|\theta) + \frac{1}{M}\sum_{m=1}^{M}\sum_{c=1}^{K} f_{m_c}(x|\theta_m)\log f_{m_c}(x|\theta_m). \tag{26}$$

Here, $f_{m_c}(x|\theta_m)$ is the output of ensemble member m for the class c on input x given the model parameters θ_m.

A second approach to measure the uncertainty in a deep ensemble is to simply look at the variance of the prediction between the different ensemble members. This captures how much the ensemble members disagree with each other.

4.3 Post-hoc Uncertainty Quantification Methods

Post-hoc methods are methods that are applied after the classification network has been trained. They are therefore computationally less expensive compared to Bayesian Neural Networks or Deep Ensemble, as post-hoc methods do not require any retraining of the classification network. In recent years, many post-hoc OOD detection methods have been introduced, with many of them being targeted for specific applications. While many of these methods were introduced as OOD detection methods, they can also be used for quantifying uncertainties in ID data. In the scope of this work, a small selection of post-hoc OOD detection and post-hoc UQ methods has been tested and compared.

4.3.1 Softmax Output

The softmax output is the output generated through the softmax activation function after the last layer of a neural network. Each class has one value, with all of them adding up to one, meaning that the softmax output could be interpreted as probabilities. This is the output that the final classification is based on, where the predicted class will be the class with the highest predicted probability.

Since the softmax activation function outputs probabilities, they can be compared between samples. A high softmax score should in theory imply a high internal model confidence. This has previously been used for OOD detection [59]. However, since the values always need to add up to one, this measure is not the most reliable. While it is the easiest measure to look at, it has been shown that the softmax output of modern neural networks is not well calibrated [60]

and therefore is not a very reliable method for measuring the model's confidence. Furthermore, there have been several studies implying that the a high softmax confidence might even not necessarily have an impact on the correctness of the prediction at all [61, 62]. It however remains a baseline method for OOD detection.

4.3.2 Trust Score

Trust score is an UQ method that was first introduced by Jiang et al. [63]. The method computes a trust score for each data sample, where a high trust score implies that the prediction is trustworthy and a low trust score suggests that the prediction might be wrong and therefore should not be trusted. The method uses a modified version of a k-nearest-neighbor classifier and estimates the agreement between that classifier and the classification network used for making predictions.

Step 1: Estimating α-high-density-sets In a first step, the α-high-density-set is calculated for each class. This set contains the samples from that class with the highest densities based on k-nearest neighbors, i.e. the α-fraction of samples with lowest empirical density are filtered out. This is to ensure that outliers do not influence the class density and therefore a good representation of the class is given. This filtering step has two hyperparameters, k and α, where a suitable value for α is data-dependent. For large data sets it is however recommended to skip the filtering for efficiency.

Step 2: Calculating Trust Score The second step is to calculate the trust score of a sample based on the α-high-density-sets obtained in the first step. The trust score is the ratio of the distance of a sample to the α-high-density-set of nearest class different from the predicted class, and the distance of that sample to the α-high-density-set of the predicted class. For a classifier h with classes Y and a test sample x, the trust score is mathematically defined as:

$$\xi(h, x) := \frac{d(x, \hat{H}_\alpha(f_{\tilde{h}(x)}))}{d(x, \hat{H}_\alpha(f_{h(x)}))}, \tag{27}$$

with $\tilde{h}(x) = \text{argmin}_{l \in Y, l \neq h(x)} d(x, \hat{H}_\alpha(f_l))$, where $\hat{H}_\alpha(f_l)$ is the α-high-density-set for class l, $h(x)$ is the predicted class from h for the sample x and $d(A, B)$ is the distance between A and B.

The trust score will be high if the sample is close to the predicted class and far away from the closest not-predicted class. Similarly, if the classifier predicts a label that is far off from the closest class, the trust score will indicate that the prediction might be wrong by returning a low score.

In the original paper, the effect of the data complexity and the performance of the trust score has been evaluated. The results show that the performance is better for data with low or medium-dimension feature spaces, as compared to high-dimensional feature spaces. This suggests that it might be challenging to produce good results for our data set with this method, as our image data is high-dimensional (180x180 pixels). Dimensionality reduction techniques to project the data to a low-dimensional vector might be a suitable solution to pre-process the data that is inputted in the trust score calculation.

4.3.3 Energy Score

Energy score is a post-hoc OOD detection method that was introduced by Liu et al. [64], specifically to improve the baseline method of using softmax outputs (see Section 4.3.1). The

energy score method is based on an energy-based model from LeCun et al. [65], which performs a mapping from input samples to singular scalar values (energies). The values correlate with the distribution of training data samples, meaning that OOD samples should be detectable with it.

Mathematically, the free energy function $E(x; f)$ for a data sample x and a neural network $f : \mathbb{R}^D \to \mathbb{R}^K$, mapping from inputs $x \in \mathbb{R}^D$ to K logits, is defined as:

$$E(x; f) = -T \cdot \log \sum_{i}^{K} e^{f_i(x)/T}, \tag{28}$$

where $f_i(x)$ denotes the i-th logit of $f(x)$ and T is the temperature parameter.

While this can be used directly for OOD detection, the authors state that ID and OOD samples will not be very well differentiable. In order to improve this, they furthermore suggest a fine-tuned version of Energy Scores which uses an energy-bounded learning objective for the neural network. This is to ensure that a clearer gap appears between ID and OOD samples. The suggested loss function is a weighted combination of the softmax output of the classification network and a regularization loss for the energies.

MOOD: Multi-level Out-of-Distribution Detection Multi-level out-of-distribution detection (MOOD) was first introduced by Lin et al. [66] and is an OOD detection method based on an adjusted energy score. It was invented to improve the original energy score method. The main idea is to not only use the logits from the last layer of the classification network, but to also look at logits at different parts throughout the network. The achitecture is similar to adaptive neural networks [67, 68, 69, 70, 71, 72] and has several exits in the network that will be analyzed. This is achieved by placing small classifiers in several locations of the network and using the resulting logits from all exits. The exits can be analyzed separately, where very far off OOD samples should be detected by earlier exits and more complex OOD samples should be detectable using later exits, but it is also possible to use the combined logits from all exits and weight them to classify between ID and OOD.

The energy score is high for unobserved data and low for observed data, meaning that there should be a threshold to separate between ID and OOD data. A previous comparison between energy score, softmax output and ensembles for OOD detection in POCUS breast imaging has been made Karlsson et al. [39], showing that the energy score outperforms the softmax-based approach.

5

Experiments

5.1 Classification Experiment

Before performing UQ or OOD detection, the general model architectures have been compared for the task of breast cancer classification. It is crucial to have a classifier that performs well at the prediction task before evaluating uncertainties and trying to find OOD samples. In the classification experiment, the performance of a base classifier CNN, which is the suggested method from previous studies on this data, is compared to the performance of BNNs and ensembles. The post-hoc methods are not evaluated here, since they are build on top of the base classifier and do not perform breast cancer classification themselves.

To evaluate the performances, different metrics are used. This includes the AUC for cancerous vs. non-cancerous, the accurarcy for the three-label prediction and the binary accuracy when benign and normal are grouped together into one class (non-cancerous). Additionally, the confusion matrices are evaluated to get a better understanding of where the models perform well and which classes might be more tricky to correctly classify.

5.2 Uncertainty Quantification Experiment

If a network's uncertainty measure performs well at reflecting the trustworthiness, predictions with low uncertainty values should almost always be believed and there should be some threshold for uncertainties after which we should not trust a prediction anymore. We therefore hypothesize that a network's performance should be better for data samples with low uncertainties as opposed to high uncertainties. In the first UQ experiment the uncertainties on the ID test set are studied using 20 different methods. This includes the three post-hoc methods softmax output, energy score and trust score built on top of a CNN classifier. For ensembles, 5 different uncertainties are evaluated (total, aleatoric and epistemic uncertainty with entropy, variance-based uncertainty and weighted variance-based uncertainty). These are tested for two ensembles: an average ensemble and a concatenation ensemble, as these have produced good results in the classification experiment. Furthermore, two types of Bayesian networks are evaluated: A BNN with VI and a BNN with MCD. For the first one, we evalute the total, aleatoric and epistemic uncertainties, while for the latter one we additionally also look at variance-based uncertainty.

In order to test our hypothesis, the data samples are sorted into five different equally-sized subsets, based on their uncertainty values, in order to check for performance differences for different ranges of uncertainties. This is done for all UQ methods and their performance will be compared. In order to compensate that for all post-hoc methods (softmax output, energy score, trust score) high values are good, the sorting is flipped here, so that the order represents uncertainties instead of certainties. For each subset, the accuracy and AUC are calculated. Furthermore, this is also done on a continuous scale, always including the samples

with lowest uncertainties until a certain point x, so that the accuracy / AUC progression based on uncertainty values can be measured.

Finally, further analysis is made on the best performing method. This includes looking at the distributions of the data samples in the five uncertainty-based subsets using PCA and calculating the uncertainty distribution of false and wrong predictions. Based on that, the likelihood function of a prediction being true given its uncertainty is developed and analyzed. Furthermore, the impact of where to set an uncertainty threshold is briefly analyzed for the best method.

5.3 Out-of-Distribution Detection Experiment

OOD detection is a binary classification task with the aim to separate between ID and OOD data. In the OOD detection experiment we test the different UQ methods for their performance on OOD detection for all three OOD data sets. We hypothesize that good UQ methods can be used to find OOD samples. ID samples should have low uncertainty value as opposed to OOD samples, which should have high uncertainties. It should be possible to find a threshold which separates ID and OOD samples well enough. This should be visible in all different OOD test sets.

AUC is used as the first evaluation metric to check if an UQ method is able to separate between ID and OOD. Furthermore, the false positive rate is calculated for two different thresholds for the uncertainties. FPR95 is a suggested method in [64], which is the false positive rate when 5% of the ID test data is classified at OOD. We additionally also look at the results when the threshold is at 20% and suggest this as a suitable method, as the results for the UQ experiment indicate this to be a suitable threshold for our data. We refer to this as FPR80.

All the methods that have previously been discussed for UQ are compared. For the ensembles, it does not matter which ensemble type would be used, as the OOD experiment is solely based on the uncertainties, i.e. the ensemble members, independent of how the final prediction of the ensemble will be made.

5.4 Design Specifications

All methods have been trained and evaluated on the same data in order to ensure comparability. For reproduability, the pre-processing steps of the data, as well as method-specific implementations and hyperparameters are descibed below.

5.4.1 Pre-Processing

All images were cropped and resized to 180 x 180 pixels and rescaled to range from 0 to 1. Additionally, for the training data set, augmentation was applied randomly during training, which consisted of spatial shifting, shearing, zooming and flipping. The shift range was 10% in both horizontal and vertical direction. The shear range was 0.1 and zoom range also 0.1. Horizontal flipping was performed with a probability of 0.5. New pixels that were outside of the boundaries of the original image where filled with the pixel value of the nearest colored pixel.

5.4.2 Base Classification Network

The base classification network used in this work is a CNN with five convolutional blocks and two dense blocks. The architecture can be seen in Figure 21, and is following the suggested

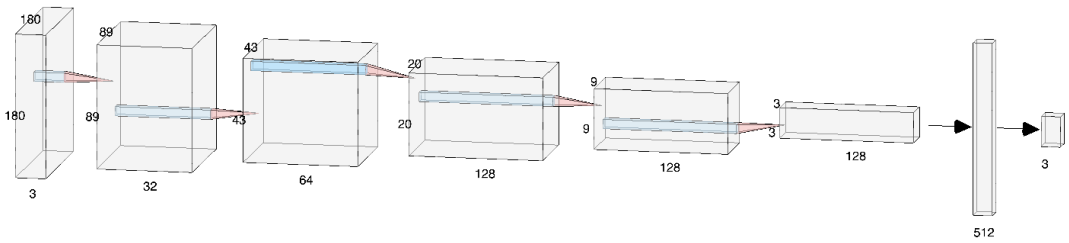

Figure 21 Architecture of the base classification network [4].

architecture in [4] based on the best performing network from Sahlin [73] with small adjustments to prevent overfitting.

Each convolutional block consists of a convolutional layer with kernel size 3×3, a ReLU activation function, a dropout of 20% and a max pooling layer with kernel size 2×2 and stride 2×2. The number of kernels for the convolutional layer in each convolutional block where 32, 64, 128, 128 and 128 respectively. After the last convolutional block, a 50% dropout was performed, then the output was flattened before the first dense block. Each dense block consists of a fully connected layer and an activation function. The first block has the size 512 and uses a ReLU activation function and 50% dropout, while the second block has the size 3 (for the classes benign, malignant and normal) and a softmax activation function. All dropouts were only used during training and deactivated for inference. The input size of the network is 180x180.

For training the network, batch size 32 was used. The learning rate was set to 0.0001 and the network was trained for 50 epochs. Adam optimizer was used. To overcome the imbalanced data set, class weights were used to balance the classes.

5.4.3 Bayesian Neural Networks

Two types of BNNs were implemented: a BNN with VI and a BNN with MCD.

BNN with Variational Inference The Bayesian Networks with VI were implemented following the guide from Shridhar et al. [29]. The posterior was solved using variational inference with Bayes by Backprop [41]. The network architecture was kept the same as for the CNN to ensure comparability, except for leaving out the dropout layers. The final BNN was trained for 100 epochs with learning rate 0.00001. During training, each image was predicted 15 times to approximate the underlying weight distributions. During inference, each test image label was predicted 25 times. The batch size was 32 and the same augmentation was used as for the CNN.

Gaussian distributions were used for the posterior and prior. The initial parameters before training were set the same for each layer. For the prior, the mean was set at 0 and the variance as 0.1.

BNN with Monte Carlo Dropout The Bayesian Network with Monte Carlo Dropout was the same as the base classification network, except that the dropout layers were also enabled during inference. Each test sample was predicted 20 times and the classification was made with the average prediction. Four different methods were implemented to calculate the uncertainties in the predictions: Total predicitive uncertainty, aleatoric and epistemic uncertainty using entropy, and the variance within the predictions. For the variance-based uncertainty, the standard deviation was calculated for each of the three classes with the 20 prediction, which were then summed up. For the entropy-based uncertainties, the ensemble equations Eq. 23, Eq. 26 and Eq. 25 were used, since the MCD method is essentially a type of ensemble rather than a BNN with weight distributions.

Table 4 Training Configurations for the CNNs within the ensemble. The parameters were chosen randomly for each of the 20 ensemble members.

Parameter	Options
Random training split	0-15%
Learning rate	0.0001-0.001
Optimizer	Adam / RMSprop
Batch size	8 / 16 / 32 / 64 / 128
Epochs	25-85

5.4.4 Neural Network Ensembles

The ensembles used in this study have been designed based on the base classification network architecture described in Section 5.4.2. In total, 20 different networks were trained and were later combined to form an ensemble. To ensure the diversification of the single networks, slightly different training settings were used for each one. Table 4 shows the range of parameters used for training. Bagging without replacement was used: For each ensemble member, random 0-15% of the training data were split away and left out, leaving the model a subset of size 85-100% of the original data set to train on. The learning rate was between 0.0001 and 0.001 and the optimizer was either Adam or RMSprop. The batch size was 8, 16, 32, 64 or 128 and the number of epochs was between 25 and 85.

Four different types of ensembles were implemented for the classification task: an average ensemble, a weighted average ensemble, a plurality vote ensemble and a concatenation ensemble. The average ensemble was calculated with the average softmax probability per class. For the weighted average ensemble, each member's output was weighted according to the average distance between the model's prediction and the average prediction of all models, making prediction further away from the mean count less towards the final prediction. The plurality vote ensemble's final prediction was calculated as the label that the most ensemble members voted for. For the concatenation ensemble, a small classification network on top of the ensemble members was trained. For an input image, the concatenation ensemble first predicts the softmax probabilities for all classes with all ensemble members and then concatenates them into a layer. After that, there is one dense layer with three outputs (benign, malignant, normal). The concatenation ensemble was trained for 40 epochs with a learning rate of 0.001, Adam optimizer and categorical cross-entropy loss with balanced class weights.

In order to calculate the uncertainties of the ensembles' predictions, five different methods were implemented and compared: variance-based uncertainty, weighted variance-based uncertainty, total uncertainty using entropy, aleatoric uncertainty using entropy and epistemic uncertainty using entropy. All of these methods are directly based on the ensemble members and can be used regardless of how the final prediction is made. The variance-based uncertainty is calculated as the sum of the variance for all three classes given the softmax outputs from the 20 ensemble members. For the weighted version, the variance for each class is weighted by the mean of that class before summation, making the classes with lower prediction probability count less. The entropy-based aleatoric and epistemic uncertainties were calculated using Eq. 25 and 26, and the total uncertainty using Eq. 23.

5.4.5 Softmax-based Method

The softmax-based method uses the classification network described in Section 5.4.2. The softmax score for each label is obtained from the raw outputs of the network.

5.4.6 Trust Score

Following the open source implementation from the original authors, the Trust Score method was implemented based on Eq. 27. As suggested in [63], we used $k = 10$ nearest neighbors for the Trust Score model. Due to data complexity and to save computational cost, the filtering parameter α was set to zero.

Several ways for inputting the data have been tested, including flattening the images and using low-dimensional feature representations. For the final implementation used in the scope of this work, we decided to use reduced-dimensional feature vectors as inputs due to the large image size. To achieve this, the logits from the first dense layer of the base classification network where used. These image representations are feature vectors of the length 512 (as opposed to the raw data with dimension 32 400).

5.4.7 Energy Score

The Energy Score method has previously been implemented by Karlsson et al. [39] and is reused in the present work. The implementation is based on Eq. 28, with the temperature $T = 0.001$. Following the suggested MOOD design, three different exits were added to the base classification network. While the last exit uses the output from the last layer of the network, the first two exits additionally use a small classifier. In both cases it consists of a convolutional layer with 128 kernels of size 3x3, followed by a ReLU activation function, a max pooling layer with kernel size 2x2, and finally a fully connected layer with three outputs. Figure 22 shows the modified architecture with the exact locations where the exits have been placed.

For both the UQ and the OOD detection experiments, the outputs from the three exits were combined. In order to determine the subsets for the UQ experiment, the energy scores were sorted for each exit separately. The final order was based on all three exits with equal voting rights on where a sample should be. As a way of achieving that, the uncertainties were sorted for each exit separately, followed by assigning a samples index in the sorted list according to its energy score as its relative UQ score from that exit. The final score of a sample was the mean relative UQ score of all three exits. This then served as the base for determining the subsets during the UQ experiment.

For the OOD detection experiment, a sample was detected as OOD if at least one of the three exits would label it as OOD. It is only labeled as ID if it passes all exits above their respective threshold. The 95% and 80% thresholds cannot be determined in a straight-forward way due to having three energy scores and the three-member voting instead of a single score. For the purpose of finding thresholds so that 95% (80%) of the ID data are labeled as ID, a suitable threshold value has to be found for each exit so that the combination of exits with the voting leads to the respective wished outcome. The implementation of this followed the suggested method by Karlsson et al. [39], which is aiming to find a threshold that cuts the same amount of data sample of for each exit. This ensures that all exits will have the same amount of times where they would detect something as OOD (most likely not always for the same samples), and that not one exit is used more than the others. To determine where to set the threshold, 200 different values were tried to find one that will lead to the desired 95% (80%) true positive rate for ID data.

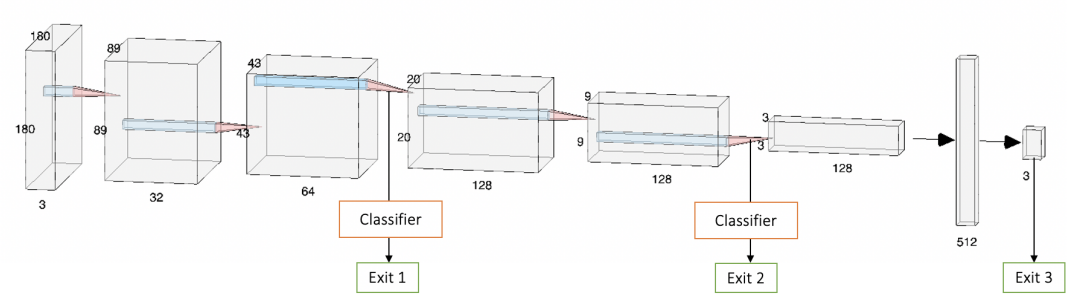

Figure 22 Architecture of CNN for the Energy Score method [39]. The CNN contains three exits, where the first two exits have additional classifiers, while the third exit uses the output from the last layer.

6

Results

This chapter demonstrates the results from the different experiments. The results are presented in the same order as the experiments.

6.1 Breast Cancer Classification

The performance results for the breast cancer classification task are shown in Table 5. The concatenation ensemble achieved the highest accuracy score (76.1%), as well as the highest binary accuracy score (84.1%). The accuracy results for the other ensemble types are similar to the ones from the CNN. The highest AUC score was achieved with the average ensemble and the plurality vote ensemble (95.6%), but between these two the average ensemble got better accuracy scores. The worst performance was achieved by the BNN with VI, with a AUC of just 61.8%, accuracy of 44.3% and binary accuracy of 62.2%, which is significantly worse than all other methods.

Figure 23 displays the confusion matrices of the predictions for the different models. The average ensemble, weighted average ensemble and plurality vote ensemble all perform similarly to each other and slightly better than the CNN. Since these three are so similar, we will only continue with one of them for further experiments. We chose the average ensemble, since it is the simplest of them. The concatenation ensemble yields the best results and will therefore also be used further on.

Table 5 The different methods' results (AUC, accuracy and binary accuracy) for classification of cancer versus non-cancer.

Method	AUC (%) ↑	ACC (%) ↑	binaryACC (%) ↑
CNN	93.3	68.6	80.3
Bayesian VI	61.8	44.3	62.2
Bayesian MCD	90.3	70.1	78.1
Average Ensemble	95.6	68.6	81.9
Weighted Average Ensemble	95.5	68.3	81.2
Plurality Vote Ensemble	95.6	68.0	80.9
Concatenation Ensemble	92.7	76.1	84.1

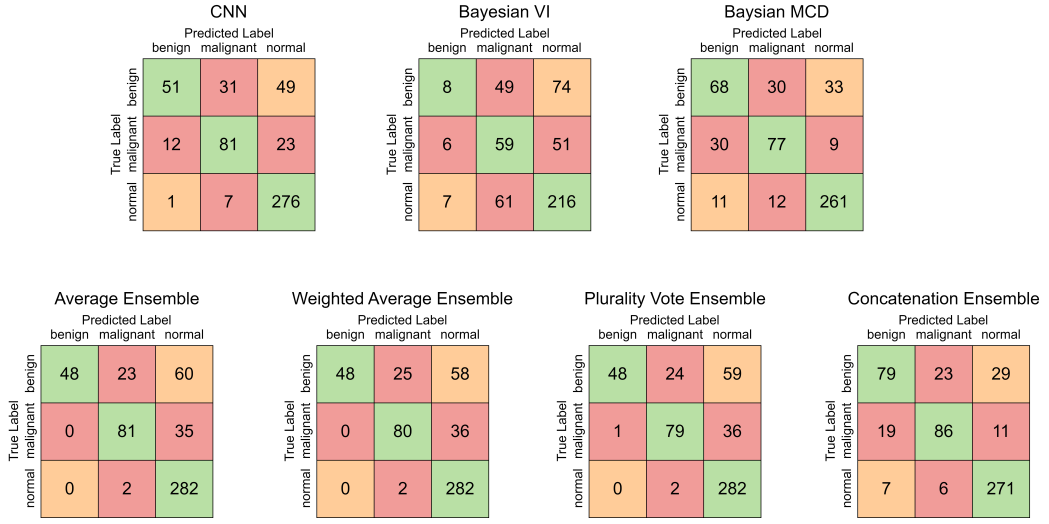

Figure 23 Confusion matrix of predicted and true labels for breast cancer classification using different models. Green represents correct prediction and red and orange represent wrong predictions, with orange being less severe than red (either missed true malignant or wrongly predicted malignant).

Table 6 Balanced accuracy on subsets of data sorted by their UQ score. Low uncertainty scores mean high certainty that the prediction is correct.

	Classifier	UQ Method	0-20%	20-40%	40-60%	60-80%	80-100%
post-hoc	CNN	Softmax	100.0	69.5	68.6	59.9	48.1
		Energy Score	90.3	78.2	63.4	61.5	54.8
		Trust Score	66.2	63.4	62.3	67.3	73.6
Ensemble	Average Ensemble	entropy, $\mathcal{U}_{tot}$	100.0	77.8	72.9	69.6	42.6
		entropy, $\mathcal{U}_{al}$	100.0	76.0	80.0	62.1	51.9
		entropy, $\mathcal{U}_{ep}$	92.8	63.0	67.1	63.7	61.2
		std	100.0	70.4	70.6	65.6	57.3
		std & weights	100.0	72.2	66.3	64.7	60.0
	Concatenation Ensemble	entropy, $\mathcal{U}_{tot}$	100.0	77.8	72.9	67.5	52.5
		entropy, $\mathcal{U}_{al}$	100.0	76.0	80.0	61.9	54.5
		entropy, $\mathcal{U}_{ep}$	92.8	63.0	63.4	63.3	68.0
		std	100.0	70.4	70.6	64.3	64.0
		std & weights	100.0	72.2	64.2	64.0	66.0
Bayesian	BCNN	VI, $\mathcal{U}_{tot}$	42.5	44.7	46.2	37.8	42.2
		VI, $\mathcal{U}_{al}$	42.5	44.7	46.2	37.8	42.2
		VI, $\mathcal{U}_{ep}$	44.0	50.1	42.6	45.0	40.8
	CNN with dropout	MCD, $\mathcal{U}_{tot}$	95.8	70.5	68.1	56.8	53.4
		MCD, $\mathcal{U}_{al}$	95.8	65.4	67.5	62.9	51.9
		MCD, $\mathcal{U}_{ep}$	94.4	70.0	64.3	68.4	55.9
		MCD, std	94.4	68.9	68.2	61.9	54.9

6.2 Uncertainty Quantification

The results on the different subsets sorted by uncertainty values are shown in Table 6 (accuracy) and Table 7 (AUC). Almost all methods that have been tested show a correlation between the uncertainties and the performance of the classification. This can be seen through the better classification performance on samples with lower uncertainties. Specifically the ensembles (both average and concatenation) using the entropy-based total uncertainty and the variance-based uncertainty are performing amongst the best. The BNN with VI achieved the worst results, followed by the post-hoc methods.

Figure 24 shows the relation of accuracy and possible threshold values for the uncertainty. The results were obatined using the average ensemble with entropy-based total uncertainty, as this seemed to be our best performing method. For a perfect UQ performance, the curve would be at 100% for most of the time and would only drop down towards the end, for the samples with the highest uncertainties. Almost all methods show a visible trend of this, with the ensemble

Table 7 AUC on subsets of data sorted by their UQ score. Low uncertainty scores mean high certainty that the prediction is correct.

	Classifier	UQ Method	0-20%	20-40%	40-60%	60-80%	80-100%
post-hoc	CNN	Softmax	100.0	99.8	96.0	82.2	75.7
		Energy Score	100.0	96.6	94.8	91.2	74.6
		Trust Score	95.0	96.5	94.0	94.3	88.5
Ensemble	Average Ensemble	entropy, $\mathcal{U}_{tot}$	100.0	99.0	96.8	95.3	64.4
		entropy, $\mathcal{U}_{al}$	100.0	98.9	97.8	94.8	68.1
		entropy, $\mathcal{U}_{ep}$	100.0	99.7	96.1	85.6	85.2
		std	100.0	100.0	95.9	86.6	57.3
		std & weights	100.0	100.0	95.4	86.5	83.8
	Concatenation Ensemble	entropy, $\mathcal{U}_{tot}$	100.0	99.1	90.0	88.3	64.7
		entropy, $\mathcal{U}_{al}$	100.0	94.3	95.9	89.0	67.8
		entropy, $\mathcal{U}_{ep}$	100.0	99.7	95.4	80.6	85.6
		std	100.0	100.0	94.9	80.0	79.6
		std & weights	100.0	100.0	94.2	82.2	83.8
Bayesian	BCNN	VI, $\mathcal{U}_{tot}$	67.8	64.6	59.0	53.6	50.2
		VI, $\mathcal{U}_{al}$	67.8	64.6	59.0	53.6	50.2
		VI, $\mathcal{U}_{ep}$	55.7	64.7	64.6	61.4	59.4
	CNN with dropout	MCD, $\mathcal{U}_{tot}$	98.0	95.3	89.3	82.5	76.0
		MCD, $\mathcal{U}_{al}$	98.0	94.6	92.5	83.2	75.6
		MCD, $\mathcal{U}_{ep}$	95.4	90.9	94.4	78.7	85.1
		MCD, std	96.2	94.7	88.2	78.8	84.9

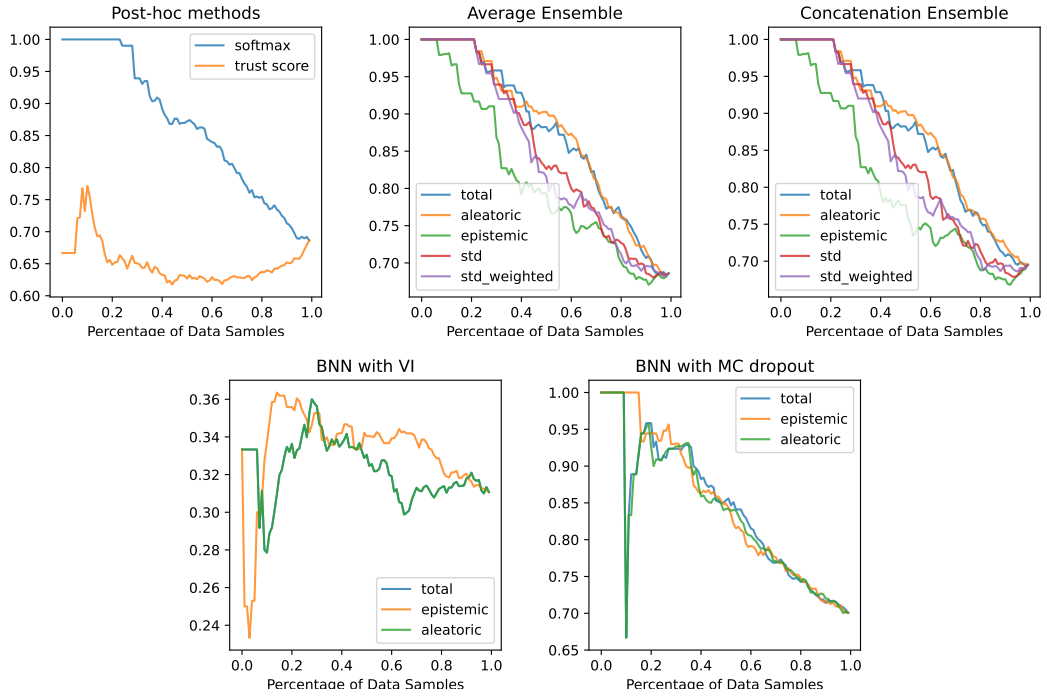

Figure 24 Accuracies by uncertainty scores. The vertical axis represents the balanced accuracy scores as function of the ratio of data samples included, where the data samples are sorted by uncertainty.

methods performing the best. A rather steep decrease in accuracy can specifically be observed when including the last 20% of data samples with the highest uncertainties. The impact of how much data to leave out, i.e. where an uncertainty threshold for trustworthiness would be, is studied in Table 8. When leaving out the 20% of samples with the highest uncertainties, the accuracy for the average ensemble increases from previous 68.6% to 77.5%, binary accuracy from 81.9% to 90.2% and the AUC from 95.6% to 98.4%. If one third of the data is left out, which is the size of data that gets misclassified, the accuracy becomes 84.5%, binary accuracy 96.1% and the AUC 99.0%.

Table 8 Classification performance when leaving out different amounts of data. The results are based on an average ensemble with entropy-based total uncertainty.

Data left out (%)	AUC (%) ↑	ACC (%) ↑	binaryACC (%) ↑
0	95.6	68.6	81.9
10	97.0	71.1	85.2
20	98.4	77.5	90.2
30	98.9	82.3	94.4
33	99.0	84.5	96.1
50	99.6	88.2	97.9

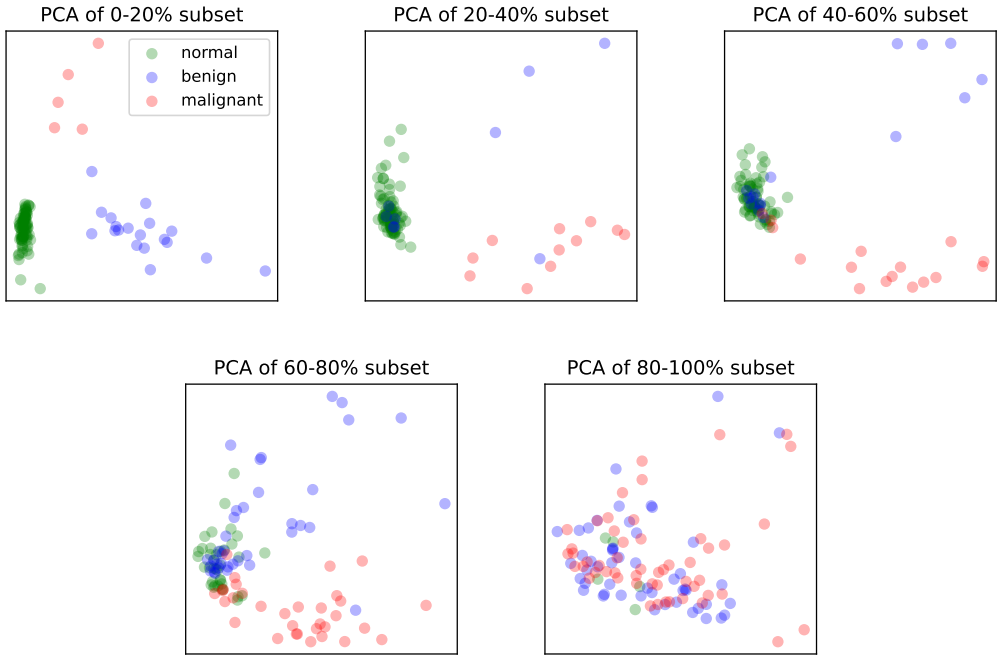

Figure 25 PCA of different subsets of the test data sorted by their uncertainties. The uncertainties were calculated with an average ensemble using the total uncertainty given by entropy as the UQ method. The PCA was performed on the feature space representations from the first dense layer in the base CNN of the respective POCUS images.

One of the best performing methods according to the results in Tables 6 and 7 is using the total uncertainty measured using entropy in an average ensemble. The different subsets using this method have been projected into a two-dimensional space using PCA and are shown in Figure 25. The 20% of samples with the lowest uncertainties can be separated really easily. The higher the uncertainties get, the harder is it also to separate the classes, with the last 20% of samples not showing any pattern of clusters. Corresponding images for the subsets with the lowest and highest uncertainties are shown in Figures 26 and 27. The images in the subset with the lowest uncertainties are visibly easy to classify using the criteria described in Section 2.2.1, as compared to the images from the subset with the highest uncertainties, which contain tissue structures that can not as easily be interpreted.

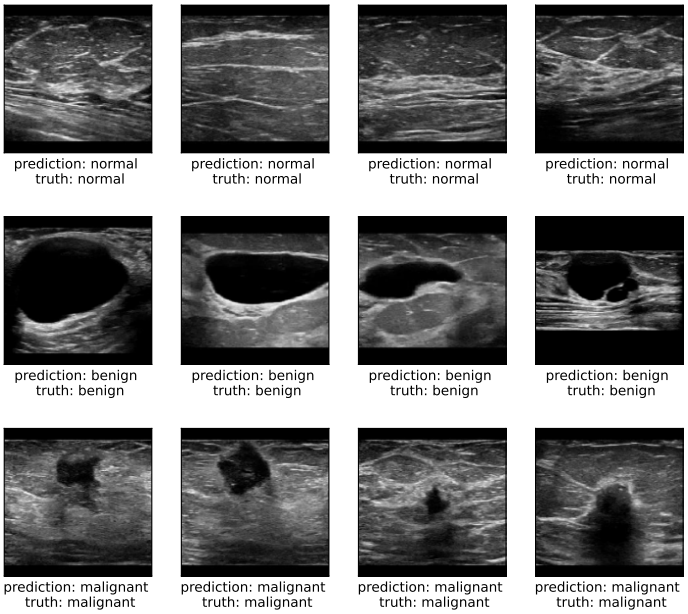

Figure 26 Images from the subset with the lowest uncertainties, using an average ensemble with the entropy-based total uncertainty.

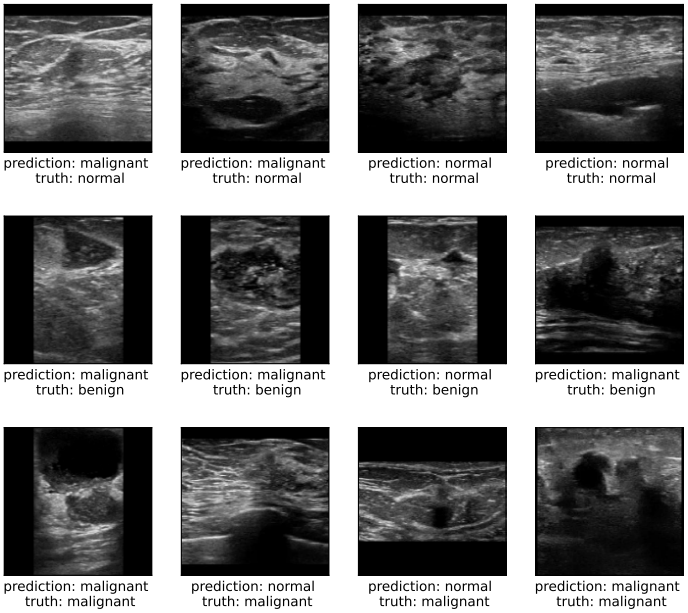

Figure 27 Images from the subset with the highest uncertainties, using an average ensemble with the entropy-based total uncertainty.

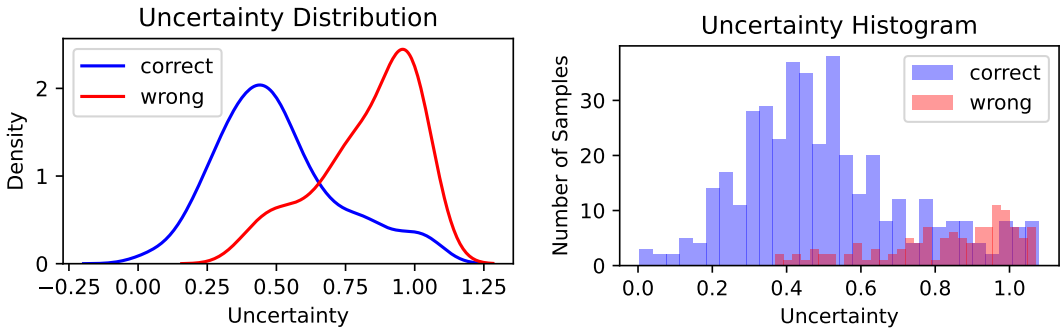

Figure 28 Uncertainty histograms and corresponding distributions for correctly and wrongly predicted test samples. The predictions were made with an average ensemble and the uncertainties were calculated using entropy (total uncertainty).

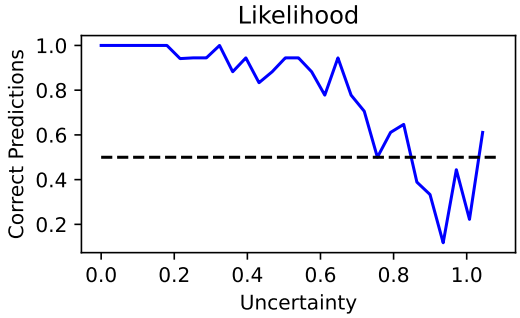

Figure 29 Likelihood of the prediction being correct given the total uncertainty calculated with entropy for an average ensemble. The function was calculated as the fraction of correct predictions over all predictions with 30 equally spaced uncertainty intervals. The horizontal line marks chance level where 50% of predictions are classified correctly.

Continuing with the same method, Figure 28 shows the distribution and histogram of uncertainties for correctly and wrongly predicted samples. While there is an overlap, making it impossible to fully separate correct from wrong predictions, there is a clear difference in the mean of both distributions. Generally, correct predictions have lower uncertainties compared to wrong predictions.

Based on the histogram, a likelihood function was investigated that plots the fraction of correct predictions based on the uncertainty values. Here, the uncertainties have been grouped together into 30 equally spaced intervals. The likelihood plot is shown in Figure 29, and shows that the likelihood of a prediction being correct is high for low uncertainties. There is a steep drop in plot towards the end, with the likelihood being below 50% for roughly the uncertainty values in the upper 20% value range. Note that this is based on the raw uncertainty values and not on the subsets as in Tables 6 and 7. The steep decrease is in alignment with the previous results, outlining that the classification performance is better for lower uncertainties.

Table 9 AUC and FPR for the different OOD detection methods evaluated on the OOD data sets. Here ↓ implies smaller values are superior and ↑ implies larger values are superior.

Method	OOD data	AUC (%) ↑	FPR95 (%) ↓	FPR80 (%) ↓
Softmax	MNIST	0.4	100.0	100.0
	CorruptPOCUS	78.1	99.4	40.4
	CCA	36.6	100.0	82.1
Energy Score	MNIST	99.4	0.0	0.0
	CorruptPOCUS	85.6	21.1	18.1
	CCA	77.8	69.0	34.5
Trust Score	MNIST	26.9	100.0	97.7
	CorruptPOCUS	55.9	98.7	88.7
	CCA	68.9	91.7	51.2
Ensemble with entropy, total uncertainty	MNIST	26.4	100.0	98.9
	CorruptPOCUS	86.4	84.4	1.3
	CCA	60.2	100.0	61.9
Ensemble with entropy, aleatoric uncertainty	MNIST	11.1	100.0	100.0
	CorruptPOCUS	35.1	94.7	89.5
	CCA	58.5	97.6	57.1
Ensemble with entropy, epistemic uncertainty	MNIST	84.6	55.2	34.5
	CorruptPOCUS	97.4	8.3	5.3
	CCA	70.2	90.5	75.0
Ensemble with std	MNIST	74.5	72.4	51.9
	CorruptPOCUS	98.1	7.9	4.5
	CCA	69.3	95.2	77.4
Ensemble with std & weights	MNIST	81.1	56.0	41.9
	CorruptPOCUS	97.2	9.6	7.0
	CCA	71.3	91.7	72.6
Bayesian VI total uncertainty	MNIST	43.6	95.3	82.2
	CorruptedPOCUS	56.0	94.2	78.5
	CCA	75.1	76.2	42.9
Bayesian VI aleatoric uncertainty	MNIST	43.6	95.3	82.2
	CorruptedPOCUS	56.0	94.2	78.5
	CCA	75.1	76.2	42.9
Bayesian VI epistemic uncertainty	MNIST	50.9	92.7	77.7
	CorruptedPOCUS	49.1	93.8	80.0
	CCA	43.4	98.8	90.5
Bayesian MCD total uncertainty	MNIST	0.5	100.0	100.0
	CorruptedPOCUS	43.8	95.2	81.0
	CCA	44.4	95.2	79.8
Bayesian MCD aleatoric uncertainty	MNIST	0.5	100.0	100.0
	CorruptedPOCUS	43.4	96.4	81.0
	CCA	43.9	95.2	83.3
Bayesian MCD epistemic uncertainty	MNIST	1.0	100.0	99.8
	CorruptedPOCUS	46.4	91.7	81.0
	CCA	46.0	91.7	77.4
Bayesian MCD std uncertainty	MNIST	0.8	100.0	100.0
	CorruptedPOCUS	45.7	91.7	78.6
	CCA	45.5	91.7	79.8

6.3 Out-of-Distribution Detection

Table 9 displays the result for the OOD detection experiment. For each method, the AUC, FPR95 and FPR80 were calculated for each of the OOD data sets. Most methods are not performing well. Energy score is the only method that performs well on all three data sets, with an FPR80 of 0% for MNIST, 18.1% for CorruptPOCUS and 34.5% for CCA, but the ensemble method outperform the energy score on the CorruptPOCUS data set (1.3% FPR80 using entropy-based total uncertainty). The BNN with VI and total uncertainty and aleatoric uncertainty performs notably well on the CCA data set (42.9% FPR80), which most methods struggle with the most detecting. The uncertainty distributions for each method and each data set can be found in Appendix B.2.

7

Discussion

7.1 Performance

In the first step, the performance of the individual methods for UQ and OOD detection, as well as the classification task, is discussed. Afterwards, the methods are compared.

7.1.1 Breast Cancer Classification

In order to make safe assessments about predictive uncertainties, the first step is to train a well-performing classifier. All methods show that the class normal is easiest to correctly classify (Figure 23). The hardest class to correctly predict is benign, with almost all classifiers predicting more than half of the benign images wrongly. The fact that this class is hard to predict seems intuitive, since benign lesion can sometimes look like malignant tumors or can be so small that it almost looks like normal tissue. Most methods perform good at finding the malignant images, however when they misclassify them, most classifiers mistake them for normal tissue more often than for benign lesions. One explanation for that can be that benign lesions are usually well circumscribed with clear borders and malignant lesions can look more blurry and irregular in shape. This can sometimes look similar to images of normal tissue of bad quality, or images of fatty tissue that is surrounded by glandular tissue.

The ensemble methods perform slightly better than the basic CNN, see Table 5. The AUC is the highest for the average ensemble and the plurality vote ensemble with 95.6%. However only the concatenation ensemble shows a noticeable improvement visible in the accuracy and binary accuracy. Compared to the CNN, the accuracy has improved to 76.1% opposed to previous 68.6%, and the binary accuracy to 84.1% from 80.3%. This observation of the superior performance of ensemble methods is in alignment with the theory of them being able to capture underlying data structures better due to using several networks. The results indicate that the ensemble members have learned to solve the problem differently, becoming proficient in slightly different areas of the input space. One possible explanation for why the concatenation ensemble performs the best is that it can learn how to weight the outputs from the ensemble members, and also how much to weigh their outputs into the predicted value for each class. This means that if one ensemble member is very good at detecting one class, but bad at the other two classes, the small network after the concatenation layer can learn to weigh the output's connections accordingly.

The BNN with VI achieved results significantly worse than all other network architectures, with an AUC of 61.8% as opposed to all other methods achieving more than 90%. From the confusion matrix shown in Figure 23, one can see that the network performed fairly good at classifying normal images, with predicting roughly 76% of them correctly. Malignant samples are classified correctly at about chance level. Almost all benign images were wrongly classified. We could not identify the reason for the poor performance in the scope of this work, but expect

it to be due to Bayesian training settings not being chosen well, more specifically the priors. Several different priors have been tested and compared. However no good setting producing good results was found. As stated in [74, 75], finding a suitable prior is crucial for the BNN's successful performance. If a good prior is used, the results of the BNN should be similar to the one of a CNN of the same architecture.

7.1.2 Uncertainty Quantification

The results of the UQ experiment are in alignment with the hypothesis that uncertainties correlate with network performance, with all methods in question having better classification results for data samples with lower uncertainties (see Tables 6, 7 and Figure 24). This correlation is especially strong for all ensemble methods, but also strong for BNNs with MCD and for the softmax output. The best results were achieved with an average ensemble with entropy-based total uncertainty. Forming different subgroups of samples based on their uncertainties shows that the three classes are easier to differentiate in samples with lower uncertainties. This explains the present correlation between uncertainties and correctness of prediction, as the classes become less and less separable for higher uncertainty scores.

The question arises on where to set a potential uncertainty threshold that determines whether a prediction is trustworthy or not. Different values were investigated that would flag between 0 and 50% of the test data set as not trustworthy, and their impact on the AUC, accuracy and binary accuracy was studied based on the best performing model (Table 8). This opens up the question for the trade-off between wanting a high performance on the samples deemed trustworthy, minimizing potential errors with severe consequences, but at the same time wanting as few samples as possible to be flagged as not trustworthy. Having a threshold value that is very low will result in the need of manual diagnosis of many images, which is precisely one of the main things this project aims to avoid. Really good results were achieved when setting the threshold so that 33% of the data was left out (deemed not trustworthy). Decreasing the threshold even more (i.e. increasing the data left out) only shows small improvements, which we believe is not reason enough to justify the amount of work that would fall into manual diagnosis without a pre-classification. We chose to continue with a threshold of leaving out 20% of data for the next experiment, since a clear drop in accuracy was visible here for most methods, but want to acknowledge that a final threshold should be chosen carefully in collaboration with experts in the area where the project is to be applied. Another solution would be to not only have one threshold, but rather use thresholds as we did in our subgrouping for the UQ experiment. This could be implemented in a way that a pre-classification is made by our algorithm and presented together with a trustworthiness score on a scale from for example one to ten, that serves as a base for an expert to look at and make the final diagnosis.

In Figures 26 and 27 we show examples of images from the subsets with the highest and lowest uncertainties based on our best performing method, with four images per label in each set. Upon visual inspection we can see that the images with the lower uncertainties are very easy to classify, portraying typical characteristics for their type of lesion, or just tissue in case of class normal. Given the rules for how to determine feature expressions of typical lesion characteristics described in Section 2.2.1, we believe that non-experts could make confident assessments and correct predictions. The images in the subset with the highest uncertainties however are less intuitive to interpret. The depicted biological structures are more blurry and messy, in addition to more artifacts being present like shadows and dark spots. While the depicted lesions might be overall harder to classify, we also believe that the image quality plays a vital role, both for human diagnosis and for an automated one using our classifiers. Non-experts would most

likely not be able to make confident assessments on these images, but experts can. This leads to the belief that with sufficient data, a neural network could also learn to classify these types of images correctly.

For the best performing method we furthermore looked into the distribution and histogram of uncertainties. The results are again in alignment with our hypothesis that there is a correlation between high uncertainties and wrong predictions. From Figure 28 we can see that it is impossible to fully separate wrong from correct predictions solely on their uncertainties. The ratio of correct to wrong predictions however decreases for higher uncertainties. This is plotted in Figure 29 and shows that the predictions are very likely to be true for smaller and medium uncertainties, but rather unlikely to be true for high uncertainties. This is again in alignment with our hypothesis.

Overall, the results from the UQ experiment serve as a proof of concept for the usefulness of UQ methods as a measure of trustworthiness in a classifier.

7.1.3 Out-of-Distribution Detection

The previous experiment's results suggest that a 5% leave-out rate for OOD detection might not be suitable in our case, and a higher value should be chosen instead. This is based on the observation that the classification performance already gets a lot worse earlier on. One could interpret this as the data samples for which high uncertainties are produced and that are also likely to be wrongly classified containing data structures that the network has not learned during training how to process and properly interpret. These samples could therefore potentially be OOD, or from a part of the distribution that was just not dense enough in the training data set for the model to learn from it accordingly. While these are just interpretations, the fact remains that the network performs bad at classifying those samples correctly, and it is therefore reasonable to have them detected by a lower threshold. As an addition to the FPR95 as a performance measure for OOD detection, we also included the FPR80.

The results show that the energy score is the only method that performs well on all three data sets. One explanation for this could be that it is a mixed method, consisting of three different exits, essentially measuring different types of differences between a sample and the samples it saw during training (more high-level and more low-level based). The results in Appendix C support this theory, showing that the different exits are good at detecting different types of OOD data, which makes a combined method very powerful and proficient on a broader scale of possible inputs.

As already indicated in the UQ experiment, the trust score method did not perform well on this experiment either. Since the trust score is a ratio, it does not take into account if a data sample if very far off from any of the classes. This, however, might be the case for some OOD data, which might be an explanation for the poor performance. Upon further investigation (see Appendix B.1), it became obvious that our input data to the trust score method, which was the logits from the first dense layer from the CNN, might not have been a suitable choice. The results there, using logits from differently trained networks, show that the method indeed is very promising and can detect OOD samples in all three data sets fairly well, given a suitable input.

While the BNN with VI performs bad at the general breast cancer classification task, it performs fairly well on the OOD detection of the CCA data set. This indicates that while it cannot classify well, it can still tell a difference in how confident it is, meaning it must notice a difference in the data. This leads us to the belief that with a well-performing BNN with VI, the UQ and OOD detection might be even better. The BNN therefore remains a promising method

for the future.

Within the different types of ensembles, the entropy-based ensemble using the epistemic uncertainty performed generally the best at the OOD detection task, with good results for MNIST and CorruptPOCUS. CCA was better detected using the aleatoric uncertainty.

Overall, it is noticeable that the different OOD data sets vary in their difficulty to be detected. The CorruptPOCUS data set is the easiest to detect, with many methods achieving high results, as shown in Table 9. This might be due to noise being used in the creation of that data set, which is something that is not present in the POCUS training and testing data. The CCA data set was the hardest to detect as OOD, which might be due to it being too close to the ID data, also being US-based images. The MNIST data set was surprisingly hard to be recognized, with only a few methods achieving reasonable results. On this data set, the energy score method achieved perfect results, detecting all samples as OOD. One guess is that this is due to the energy score having access to three scores from different part throughout the network, where feature maps will look very different for a POCUS image compared to an MNIST image that consists of pixel values zero for the most part.

7.1.4 Method Comparison

Most of the investigated methods have shown to produce promising results that can potentially be used in a real-world setting for using deep learning to classify lesions in POCUS breast imaging. However, they have different advantages and disadvantages, meaning a suitable method has to be chosen carefully. Based on our results and comparisons to related work, we developed a comparison overview of the different methods regarding computational cost, training procedures, implementation efforts, quality of UQ, ability to decompose the uncertainty into aleatoric and epistemic, quality of OOD detection, and suitability for high dimensional data and large data sets. The overview is shown in Table 10. The ensemble methods were grouped together here, as they all fall in the same category and according to our results, the type of ensemble and type of uncertainty measure one should use depends on the specific problem and data set. Note that we include MCMC here for completion, even though it was not implemented in the scope of this work.

Ensembles, while producing some of the best results, are computationally very costly, leading to really high training times and a lot of required memory storage for model parameters compared to the other methods, and are therefore unsuitable for many applications [76], especially when the training data sets are large. However, they are very easy to implement compared to Bayesian networks and only a few modification need to be made to the implementation of a base classifier. If it was just for the purpose of the classification task, it is questionable if an ensemble should be used, as the results in Table 5 show that most ensemble method are performing only slightly better than the CNN. The only method that is noticeably better at the breast cancer classification task according to the accuracy is the concatenation ensemble.

Even though the softmax score was only introduced as a baseline metric, it achieves noticeably good results on the UQ experiment. Compared to the other post-hoc methods energy score and trust score, softmax performs the best at the UQ experiment. It is however clearly outperformed by the energy score method at the OOD detection task. One possible explanation for the good performance at the UQ experiment and not so good performance at the OOD experiment is that the test data used in the UQ experiment is very similar to the training data. For such data, the network has learned how to confidently interpret the results, with a high softmax score for the predicted class indicating that the network is very sure about this prediction. For OOD data however, there are structures present in the data that do not align

Table 10 Comparison of the different uncertainty quantification methods. These results were partly obtained in the experiments of this work, and partly compared to findings in [27] and [34].

	Bayesian Neural Networks			Ensembles	*Post-hoc UQ methods*		
	MCMC	MCD	VI		Softmax	Trust	Energy
Computational cost	very high	low	high-medium	high	very low	low	low
Requires Re-training	yes	no (if dropout layers exist)	yes	yes	no	no	no (unless re-trained on exits)
Effort to implement	very high	low	high	medium-low	very low	medium	medium
Quality of UQ	high-medium	medium-low	medium	high	low	medium-low	medium
Ability to decompose into epistemic and aleatoric	yes	yes	yes	yes	no	no	no
Quality of OOD detection	low	low	low	medium	very low	low	high-medium
Suitable for high dimensional data	no	yes	yes	yes	yes	only with dimensionality reduction	yes
Suitable for large data sets	no	yes	yes	depends on training settings and number of ensemble members	yes	yes	yes

with the rules and patterns the network learned, therefore possibly leading to a more random performance of the network on OOD data and the maximum softmax score not being as high. This belief was verified by looking at the distributions of softmax scores, as can be seen in Appendix B.

The Bayesian MCD method performed quite good at the UQ experiment, but not good at all at the OOD detection one. This observation is in alignment with the literature findings we included in Table 10.

While the energy score method was clearly the best at the OOD detection task, it was outperformed at the UQ experiment by several of the ensemble methods, with the best result coming from the average ensemble with entropy-based total uncertainty. In direct comparison, we argue that the latter might be a better choice for building safety mechanisms into a classifier for POCUS images. This is due to the UQ experiment being more important for that application than the OOD detection task. However, both methods are promising and should be investigated further.

7.1.5 Comparison with Smaller Training Data Set

A previous, smaller version of our current training data set has been used in the beginning of this project, with the testing data set staying the same. This data set was also previously used in [39] for an OOD detection experiment, where the methods softmax output, energy score, ensembles with variance-based uncertainty and ensembles with weighted variance-based uncertainty were studied and compared. To get a more complete comparison, we performed our OOD detection experiment for all methods of the previous data set. The results can be found in

Appendix A. This is to gain insights into the impact of available training data on the method's performances. The classification experiment was also repeated.

Generally, the models got better at the breast cancer classification task with the new data set. This is due to the training data set being larger and therefore the model being able to learn more general rules. The only method that was slightly better on the previous data set was the concatenation ensemble.

For the OOD detection experiment, the softmax results got a lot better, which could be due to the network being more secure in its learned rules for interpreting images. We cannot exclude the possibility of these results being somewhat random though due to the nature of the softmax score and the somewhat random performance on assigning labels to OOD data. The energy score method got better at detecting CorruptPOCUS samples with more available data, but shows no significant difference for the other two data sets. The trust score method got better on CorruptPOCUS and CCA, but due to our findings in Appendix B.1 we believe that these results should only be interpreted carefully. For the ensembles, the effect of the larger data set seems to be mixed. Some ensemble-based UQ methods got better at detecting OOD samples and some got worse, with it differing largely across the three different OOD data sets. All of them got slightly better at detecting CorruptPOCUS, however, for MNIST most ensemble methods got worse. The performance on CCA also got worse. The BNN with VI got better for almost all uncertainty measures and data sets, with only one exception. The FPR80 for CCA with the total uncertainty and aleatoric uncertainty was even halved. The results for BNNs with MCD showed no significant difference.

Overall, the increased size of training data has mixed impacts, but more positive ones than negative ones. We therefore argue that it is beneficial to use a larger training data size and believe that with even more data, the methods are likely to perform even better, both at classification and at OOD detection.

7.2 Limitations

Several limitations impact the outcome of this study and should be addressed. One big limitation is the very limited data set, specifically for POCUS images. One of the most important factors for successful training of a good classifier is the training data. With too little data, it becomes hard to generalize and achieve good results on a broad scale. Therefore, the limited amount of training data, which is additionally imbalanced, has been a major drawback for this project. We expect that with more data, classifiers for this problem would potentially perform better, and UQ methods should be more stable, since the underlying ID distribution will be known better.

Another possible problem is that the ID data sets (both training and testing) might not be perfect. Since we do not know the underlying data distribution, we cannot guarantee that there are only ID samples in our ID data set. This refers to there potentially being data outliers that should be excluded from the ID distribution. The results from the UQ experiment and the OOD detection experiment suggest that there are indeed samples that are most likely not ID, as the networks produce high uncertainties for such. This has to be taken into account when setting thresholds that will later be applied in a real-life application.

Upon further exploration of the subset of POCUS test samples with the highest uncertainties, the question arises whether all of them are in fact labeled correctly. While all images were labeled by radiologists, we cannot exclude the possibility of mistakes, especially in cases were images were labeled without biopsy results. This can introduce mistakes and, especially with

such limited data, can confuse the algorithm during training.

Furthermore, a possible bias in our POCUS and US data sets might exist due to homogeneity of patients. The collected data is mostly from white patients, which might have an impact on the images that could potentially make a difference for an algorithm. Additionally, the age groups are also imbalanced, and the age is completely left out in the classification process. The age factor however can serve as a helpful indicator for radiologists to make an assessment, and therefore should maybe also be included in the algorithmic classification procedure.

Lastly, another limitation worth mentioning is that the networks base their classification solely on one image, while radiologists have access to moving imagery. This can be a crucial factor in diagnosis. When pressure is applied to the breast tissue with the ultrasound probe, a malignant tumor will stay in place, while fibroadenomas can shift and cysts can move around. Not having access to this information possibly limited our networks in their ability to differentiate malignant and benign findings.

7.3 Future Development

This thesis serves as a baseline for UQ for breast cancer classification in POCUS imaging and shows that is possible to find thresholds that reduce the risk of believing a false prediction. For future development, more UQ methods should be explored, as none of the investigated methods shows a very good performance on the UQ experiment and the OOD detection for all three OOD data sets. Specifically Bayesian networks should be explored more, as they should produce promising results when trained correctly. Training a Bayesian network however remains a difficult task which needs to be tackled in the future.

Another method that should be explored in the future is ensemble distillation to reduce the computational cost of the ensemble methods. If this is successful, it might be employable for the final application. Furthermore, the energy score method should be explored more, since it showed promising results due to the combination of several detectors. An idea would be to try out more exits. Based on that idea of combining different OOD detectors using a voting system where a sample is labeled as OOD if at least one detector flags it as OOD, we could potentially also invent our own UQ method that combines several of the methods investigated in this thesis. Finally, active learning is also an interesting method that could be explored if this project would be applied in real-world medical settings.

How to decide on uncertainty thresholds and how to implement them into an actual application remains an open question. In the future, this should be investigated together with expert radiologists and experts from regions where such a project would be used in practise. We propose that a confidence-of-prediction score should be used instead of a binary system only deciding between trustworthy and not trustworthy. Whether this should be a score from e.g. one to ten, a percentage number, or something else, remains to be an open question outside of our expertise.

In order to improve both the classification performance and the UQ quality, more data is needed in the future. This should include first and foremost more POCUS data, however synthetic data could also be explored more for training.

An interesting extension to this work would be working with small video snippets instead of images and test if this might improve the performance. This is however computationally very costly, and might therefore not be feasible in practice.

To ensure that this project is useful and applicable in settings where access to breast cancer screening is not given, field studies should be done. A first study will be done in the near future

in Kenya and further development or changes to this work should be made based on that.

Finally, it is important to mention that in order to further advance this project, it is crucial to collaborate with projects finding solutions for the treatments after a diagnosis with our tool. Only with that, a truthfully positive impact can be made that might have the potential to save lives.

8

Conclusion

This work investigated the potential of using different uncertainty quantification methods to assess the trustworthiness of a prediction by calculating uncertainty scores. Our results show that all investigated UQ methods produce uncertainties that correlate with the likelihood of predictions being correct. This is in alignment with the initial hypothesis that UQ measures can be used as a safety concept to determine the trustworthiness of a prediction. The correlation was the strongest when using an average ensemble as the base classification method with the entropy-based total uncertainty. For our specific data, the results suggest that an uncertainty threshold should be set at a value where around 80% of test data has smaller uncertainties, cutting off the data samples with the highest 20% uncertainties. As the accuracy on that subset of data is below 50%, we suggest that a radiologist should determine the diagnosis in those cases and the prediction from our DL algorithm should not be used. For samples with uncertainties below the threshold, our predictions can serve as a pre-classification of the POCUS images.

Furthermore, only some methods succeed at detecting OOD data samples based on their uncertainty scores. The performance differs widely across the different OOD data sets, with CorruptPOCUS being the easiest to differentiate from the POCUS test data set, and CCA being the hardest. The energy score method outperforms all other methods in this experiment, being the only one confidently detecting OOD samples in all three OOD data sets. For the MNIST data set, it correctly detects all samples as OOD already at the 95% uncertainty threshold. The best performance on the CorruptPOCUS data set was achieved with an average ensemble using the variance-based uncertainty. For the CCA data set, energy score performed the best, closely followed by a BNN with VI.

The results set a baseline and show that the inclusion of UQ methods into an application for automated breast cancer classification in POCUS images has great potential for the future. Introducing uncertainty thresholds increases the accuracy of the remaining predictions and is therefore a great step towards building a trustworthy classifier. Further research needs to be done to investigate more UQ methods, as well as more data needs to be collected in order to improve network training and hopefully improve the overall classification performance.

Bibliography

[1] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray. "Global cancer statistics 2020: globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries". *CA: a cancer journal for clinicians* **71**:3 (2021), pp. 209–249.

[2] S. Shapiro, W. Venet, P. Strax, L. Venet, and R. Roeser. "Ten-to fourteen-year effect of screening on breast cancer mortality". *Journal of the National Cancer Institute* **69**:2 (1982), pp. 349–355.

[3] L. Tabar, M.-F. Yen, B. Vitak, H.-H. T. Chen, R. A. Smith, and S. W. Duffy. "Mammography service screening and mortality in breast cancer patients: 20-year follow-up before and after introduction of screening". *The Lancet* **361**:9367 (2003), pp. 1405–1410.

[4] J. Karlsson, I. Arvidsson, F. Sahlin, K. Åström, N. C. Overgaard, K. Lång, and A. Heyden. "Classification of point-of-care ultrasound in breast imaging using deep learning". In: *Medical Imaging 2023: Computer-Aided Diagnosis*. Vol. 12465. SPIE. 2023, pp. 192–200.

[5] S. K. Zhou, H. Greenspan, C. Davatzikos, J. S. Duncan, B. Van Ginneken, A. Madabhushi, J. L. Prince, D. Rueckert, and R. M. Summers. "A review of deep learning in medical imaging: imaging traits, technology trends, case studies with progress highlights, and future promises". *Proceedings of the IEEE* **109**:5 (2021), pp. 820–838.

[6] B. Lakshminarayanan, A. Pritzel, and C. Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". *Advances in neural information processing systems* **30** (2017).

[7] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. "Concrete problems in ai safety". *arXiv preprint arXiv:1606.06565* (2016).

[8] E. Hüllermeier and W. Waegeman. "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods". *Machine Learning* **110** (2021), pp. 457–506.

[9] J. P. McGahan, M. A. Pozniak, J. Cronan, J. S. Pellerito, K. S. Lee, M. Blaivas, and P. Cooperberg. "Handheld ultrasound: threat or opportunity?" *Applied Radiology* **44**:3 (2015), p. 20.

[10] L. Lee and J. M. DeCara. "Point-of-care ultrasound". *Current Cardiology Reports* **22** (2020), pp. 1–10.

[11] C. F. Dietrich, A. Goudie, L. Chiorean, X. W. Cui, O. H. Gilja, Y. Dong, J. S. Abramowicz, S. Vinayak, S. C. Westerway, C. P. Nolsøe, et al. "Point of care ultrasound: a wfumb position paper". *Ultrasound in medicine & biology* **43**:1 (2017), pp. 49–58.

[12] T. A. Reynolds, S. Amato, I. Kulola, C.-J. J. Chen, J. Mfinanga, and H. R. Sawe. "Impact of point-of-care ultrasound on clinical decision-making at an urban emergency department in tanzania". *PloS one* **13**:4 (2018), e0194774.

[13] M. A. Huson, D. Kaminstein, D. Kahn, S. Belard, P. Ganesh, V. Kandoole-Kabwere, C. Wallrauch, S. Phiri, B. Kreuels, and T. Heller. "Cardiac ultrasound in resource-limited settings (curls): towards a wider use of basic echo applications in africa". *The Ultrasound Journal* **11**:1 (2019), pp. 1–10.

[14] Y. Baribeau, A. Sharkey, O. Chaudhary, S. Krumm, H. Fatima, F. Mahmood, and R. Matyal. "Handheld point-of-care ultrasound probes: the new generation of pocus". *Journal of cardiothoracic and vascular anesthesia* **34**:11 (2020), pp. 3139–3145.

[15] J. L. Dıaz-Gómez, P. H. Mayo, and S. J. Koenig. "Point-of-care ultrasonography". *New England Journal of Medicine* **385**:17 (2021), pp. 1593–1602.

[16] N. S. El Saghir, C. A. Adebamowo, B. O. Anderson, R. W. Carlson, P. A. Bird, M. Corbex, R. A. Badwe, M. A. Bushnaq, A. Eniu, J. R. Gralow, et al. "Breast cancer management in low resource countries (lrcs): consensus statement from the breast health global initiative". *The Breast* **20** (2011), S3–S11.

[17] L. E. Pace and L. N. Shulman. "Breast cancer in sub-saharan africa: challenges and opportunities to reduce mortality". *The oncologist* **21**:6 (2016), pp. 739–744.

[18] W. Y. Joko-Fru, A. Miranda-Filho, I. Soerjomataram, M. Egue, M.-T. Akele-Akpo, G. N'da, M. Assefa, N. Buziba, A. Korir, B. Kamate, et al. "Breast cancer survival in sub-saharan africa by age, stage at diagnosis and human development index: a population-based registry study". *International journal of cancer* **146**:5 (2020), pp. 1208–1218.

[19] C. Allemani, T. Matsuda, V. Di Carlo, R. Harewood, M. Matz, M. Nikšić, A. Bonaventure, M. Valkov, C. J. Johnson, J. Estève, et al. "Global surveillance of trends in cancer survival 2000–14 (concord-3): analysis of individual records for 37 513 025 patients diagnosed with one of 18 cancers from 322 population-based registries in 71 countries". *The Lancet* **391**:10125 (2018), pp. 1023–1075.

[20] J. Okello, H. Kisembo, S. Bugeza, and M. Galukande. "Breast cancer detection using sonography in women with mammographically dense breasts". *BMC medical imaging* **14**:1 (2014), pp. 1–8.

[21] J. Karlsson and J. Ramkull. *Machine learning algorithm for classification of breast ultrasound images*. Student Paper. 2021.

[22] Smithuis, Robin and Wijers, Lidy and Dennert, Indra. "Ultrasound of the breast" (). `https://radiologyassistant.nl/breast/ultrasound/ultrasound-of-the-breast` Accessed: 2023-11-27.

[23] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[24] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". *nature* **521**:7553 (2015), pp. 436–444.

[25] A. Krenker, J. Bešter, and A. Kos. "Introduction to the artificial neural networks". *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech* (2011), pp. 1–18.

[26] K. O'Shea and R. Nash. "An introduction to convolutional neural networks". *arXiv preprint arXiv:1511.08458* (2015).

[27] V. Nemani, L. Biggio, X. Huan, Z. Hu, O. Fink, A. Tran, Y. Wang, X. Du, X. Zhang, and C. Hu. "Uncertainty quantification in machine learning for engineering design and health prognostics: a tutorial". *arXiv preprint arXiv:2305.04933* (2023).

[28] A. Der Kiureghian and O. Ditlevsen. "Aleatory or epistemic? does it matter?" *Structural safety* **31**:2 (2009), pp. 105–112.

[29] K. Shridhar, F. Laumann, and M. Liwicki. "A comprehensive guide to bayesian convolutional neural network with variational inference". *arXiv preprint arXiv:1901.02731* (2019).

[30] A. Olmin. *On Uncertainty Quantification in Neural Networks: Ensemble Distillation and Weak Supervision*. PhD thesis. Linköping University Electronic Press, 2022.

[31] A. Malinin and M. Gales. "Predictive uncertainty estimation via prior networks". *Advances in neural information processing systems* **31** (2018).

[32] K. P. Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

[33] J. D. Lee and K. A. See. "Trust in automation: designing for appropriate reliance". *Human factors* **46**:1 (2004), pp. 50–80.

[34] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al. "A review of uncertainty quantification in deep learning: techniques, applications and challenges". *Information fusion* **76** (2021), pp. 243–297.

[35] J. Yang, K. Zhou, Y. Li, and Z. Liu. "Generalized out-of-distribution detection: a survey. arxiv 2021". *arXiv preprint arXiv:2110.11334* ().

[36] GE Healthcare. "Vscan air" (). `https://vscan.rocks/product/vscanair` Accessed: 2023-11-27.

[37] GE Healthcare. "Logiq e10 ultrasound series" (). `https://www.gehealthcare.com/products/ultrasound/logiq/logiq-e10` Accessed: 2023-11-27.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* **86**:11 (1998), pp. 2278–2324.

[39] J. Karlsson, M. Wodrich, N. C. Overgaard, F. Sahlin, K. Lång, A. Heyden, and I. Arvidsson. *Towards out-of-distribution detection for breast cancer classification in point-of-care ultrasound imaging*. 2024. arXiv: `2402.18960 [cs.CV]`.

[40] Zukal, Martin and Beneš, Radek, Číka, Petr and Říha, Kamil. "Ultrasound image database" (). `http://splab.cz/en/download/databaze/ultrasound` Accessed: 2023-11-08.

[41] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. "Weight uncertainty in neural network". In: *International conference on machine learning*. PMLR. 2015, pp. 1613–1622.

[42] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equation of state calculations by fast computing machines". *The journal of chemical physics* **21**:6 (1953), pp. 1087–1092.

[43] W. K. Hastings. "Monte carlo sampling methods using markov chains and their applications" (1970).

[44] M. Betancourt. "A conceptual introduction to hamiltonian monte carlo". *arXiv preprint arXiv:1701.02434* (2017).

[45] T. Chen, E. Fox, and C. Guestrin. "Stochastic gradient hamiltonian monte carlo". In: *International conference on machine learning*. PMLR. 2014, pp. 1683–1691.

[46] G. E. Hinton and D. Van Camp. "Keeping the neural networks simple by minimizing the description length of the weights". In: *Proceedings of the sixth annual conference on Computational learning theory*. 1993, pp. 5–13.

[47] A. Graves. "Practical variational inference for neural networks". *Advances in neural information processing systems* **24** (2011).

[48] Y. Gal and Z. Ghahramani. "Dropout as a bayesian approximation: representing model uncertainty in deep learning". In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[49] F. Verdoja and V. Kyrki. "Notes on the behavior of mc dropout". *arXiv preprint arXiv:2008.02627* (2020).

[50] L. L. Folgoc, V. Baltatzis, S. Desai, A. Devaraj, S. Ellis, O. E. M. Manzanera, A. Nair, H. Qiu, J. Schnabel, and B. Glocker. "Is mc dropout bayesian?" *arXiv preprint arXiv:2110.04286* (2021).

[51] L. K. Hansen and P. Salamon. "Neural network ensembles". *IEEE transactions on pattern analysis and machine intelligence* **12**:10 (1990), pp. 993–1001.

[52] D. Opitz and R. Maclin. "Popular ensemble methods: an empirical study". *Journal of artificial intelligence research* **11** (1999), pp. 169–198.

[53] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke. "Out-of-distribution detection using an ensemble of self supervised leave-out classifiers". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 550–564.

[54] T. Han and Y.-F. Li. "Out-of-distribution detection-assisted trustworthy machinery fault diagnosis approach with uncertainty-aware deep ensembles". *Reliability Engineering & System Safety* **226** (2022), p. 108648.

[55] D. Yang, K. Mai Ngoc, I. Shin, K.-H. Lee, and M. Hwang. "Ensemble-based out-of-distribution detection". *Electronics* **10**:5 (2021), p. 567.

[56] T. G. Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[57] L. Breiman. "Bagging predictors". *Machine learning* **24** (1996), pp. 123–140.

[58] R. E. Schapire. "The strength of weak learnability". *Machine learning* **5** (1990), pp. 197–227.

[59] D. Hendrycks and K. Gimpel. "A baseline for detecting misclassified and out-of-distribution examples in neural networks". *arXiv preprint arXiv:1610.02136* (2016).

[60] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. "On calibration of modern neural networks". In: *International conference on machine learning*. PMLR. 2017, pp. 1321–1330.

[61] A. Nguyen, J. Yosinski, and J. Clune. "Deep neural networks are easily fooled: high confidence predictions for unrecognizable images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 427–436.

[62] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples". *arXiv preprint arXiv:1412.6572* (2014).

[63] H. Jiang, B. Kim, M. Guan, and M. Gupta. "To trust or not to trust a classifier". *Advances in neural information processing systems* **31** (2018).

[64] W. Liu, X. Wang, J. Owens, and Y. Li. "Energy-based out-of-distribution detection". *Advances in neural information processing systems* **33** (2020), pp. 21464–21475.

[65] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. "A tutorial on energy-based learning". *Predicting structured data* **1**:0 (2006).

[66] Z. Lin, S. D. Roy, and Y. Li. "Mood: multi-level out-of-distribution detection". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2021, pp. 15313–15323.

[67] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama. "Adaptive neural networks for efficient inference". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 527–536.

[68] G. Huang, D. Chen, T. Li, F. Wu, L. Van Der Maaten, and K. Q. Weinberger. "Multi-scale dense networks for resource efficient image classification". *arXiv preprint arXiv:1703.09844* (2017).

[69] Z. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman, and R. Feris. "Blockdrop: dynamic inference paths in residual networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8817–8826.

[70] H. Li, H. Zhang, X. Qi, R. Yang, and G. Huang. "Improved techniques for training adaptive deep networks". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1891–1900.

[71] A. Graves. "Adaptive computation time for recurrent neural networks". *arXiv preprint arXiv:1603.08983* (2016).

[72] A. Veit and S. Belongie. "Convolutional networks with adaptive inference graphs". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 3–18.

[73] F. Sahlin. *Detection of breast cancer in pocket ultrasound images using deep learning*. Student Paper. 2022.

[74] V. Fortuin. "Priors in bayesian deep learning: a review". *International Statistical Review* **90**:3 (2022), pp. 563–591.

[75] B.-H. Tran, S. Rossi, D. Milios, and M. Filippone. "All you need is a good functional prior for bayesian deep learning". *The Journal of Machine Learning Research* **23**:1 (2022), pp. 3210–3265.

[76] A. Malinin, B. Mlodozeniec, and M. Gales. "Ensemble distribution distillation". *arXiv preprint arXiv:1905.00076* (2019).

A

Results on Previous Data Set

The data set sizes of the previous data set are shown in Table 11. This data set was previously used in a study about Out-of-Distribution detection by Karlsson et al. [39]. This data set has been extended by adding 245 more POCUS images and 493 more US images to the training data set, while the test data set stayed the same.

The classification experiment and OOD detection experiment in this work have also been tested on the old data set and the results are shown below.

Table 11 Sizes of the previous ID data sets. The test set was kept the same, and more data was only added to the training set.

		POCUS	US	Total	Data set size
Train	Normal	304	168	472	1236
	Benign	140	101	241	
	Malignant	125	398	523	
Test	Normal	284	-	284	531
	Benign	131	-	131	
	Malignant	116	-	116	
Total		1100	667	1767	

A.1 Breast Cancer Classification Experiment

The results of the breast cancer classification experiment are shown in Table 12. Almost all methods performed slightly worse with less availble training data.

Table 12 The different methods' results for classification of cancer versus non-cancer trained on the previous data set.

Method	AUC (%) ↑	ACC (%) ↑	binaryACC (%) ↑
CNN	93.3	65.4	72.5
Bayesian VI	50.0	42.2	57.9
Bayesian MCD	89.2	67.3	69.9
Average Ensemble	96.6	63.4	76.0
Weighted Average Ensemble	96.5	63.1	75.8
Plurality Vote Ensemble	96.6	64.0	77.3
Concatenation Ensemble	94.9	77.1	87.2

A.2 Out-of-Distribution Detection Experiment

The results of the OOD detection experiment are shown in Table 13.

Table 13 AUC and FPR for the different OOD detection methods evaluated on the OOD data sets, with the classifiction networks being trained on the previous data set. Here ↓ implies smaller values are superior and ↑ implies larger values are superior.

Method	OOD data	AUC (%) ↑	FPR95 (%) ↓	FPR80 (%) ↓
Softmax	MNIST	3.8	100.0	100.0
	CorruptPOCUS	17.4	100.0	96.4
	CCA	28.5	100.0	96.4
Energy Score	MNIST	99.4	0.0	0.0
	CorruptPOCUS	95.5	8.7	6.4
	CCA	77.7	79.8	38.1
Trust Score	MNIST	57.1	99.7	96.1
	CorruptPOCUS	45.4	97.7	91.9
	CCA	56.1	100.0	81.0
Ensemble with entropy, total uncertainty	MNIST	35.9	100.0	100.0
	CorruptPOCUS	90.1	78.0	4.5
	CCA	77.3	97.6	53.6
Ensemble with entropy, aleatoric uncertainty	MNIST	11.7	100.0	100.0
	CorruptPOCUS	50.2	96.0	90.4
	CCA	71.7	96.4	53.6
Ensemble with entropy, epistemic uncertainty	MNIST	93.4	21.9	9.7
	CorruptPOCUS	95.1	14.5	9.2
	CCA	74.9	67.9	45.2
Ensemble with std	MNIST	87.8	43.1	18.4
	CorruptPOCUS	96.3	14.7	9.0
	CCA	81.8	76.2	39.3
Ensemble with std & weights	MNIST	92.6	20.1	11.0
	CorruptPOCUS	95.2	15.4	9.0
	CCA	81.2	65.5	36.9
Bayesian VI total uncertainty	MNIST	30.9	98.8	93.9
	CorruptedPOCUS	55.4	94.4	75.3
	CCA	48.8	95.2	85.7
Bayesian VI aleatoric uncertainty	MNIST	30.9	98.8	93.9
	CorruptedPOCUS	55.4	94.4	75.3
	CCA	48.8	95.2	85.7
Bayesian VI epistemic uncertainty	MNIST	53.2	90.4	74.1
	CorruptedPOCUS	52.6	94.5	75.5
	CCA	52.2	95.2	76.2
Bayesian MCD total uncertainty	MNIST	6.2	100.0	100.0
	CorruptedPOCUS	41.9	96.4	84.5
	CCA	42.3	96.4	83.3
Bayesian MCD aleatoric uncertainty	MNIST	4.5	100.0	100.0
	CorruptedPOCUS	40.8	98.8	86.9
	CCA	41.1	98.8	84.5
Bayesian MCD epistemic uncertainty	MNIST	20.4	92.6	87.6
	CorruptedPOCUS	48.0	89.3	77.4
	CCA	49.2	89.3	75.0
Bayesian MCD std uncertainty	MNIST	15.4	97.0	92.4
	CorruptedPOCUS	46.4	91.7	75.0
	CCA	47.1	90.5	73.8

B

Additional Evaluation Results

B.1 Trust Score

During the evaluation process, we tried different CNNs for evaluating the trust score. In the final results, we used the same CNN as in the paper that this work is based on, with learning rate 0.0001 and 50 epochs, to ensure comparability. However, we noticed that the trust score method is highly influenced by the network's representations of the data in the first dense layer, and that these seem to vary a lot for different training settings. This is shown Table 14. All models compared here produced similar results on the classification task. The results however show very different AUCs and FPRs for the OOD detection, reflecting the Trust Score's high dependence on the feature representations. This suggests that perhaps using the logits from the first dense layer in the CNN as a feature representation of the input image is not suitable, and alternative solutions would have to be explored. This could include training an autoencoder and using the latent space representations as inputs for the trust score calculation.

B.2 Uncertainty Distributions

Figure 30 shows the distributions of the uncertainties for the different test data sets using the different UQ methods. The plots include the thresholds where 95% and 80% of the ID would be correctly classified as ID. All samples with uncertainties higher than the respective threshold would be classified as OOD. In order for the softmax and trust score plots to represent uncertainties instead of certainties, the scores were negated. The results for the energy score are not included here, since there is not a single score, but three that make a combined decision for the OOD detection. Results for different exits can be seen in Appendix C.

Table 14 AUC and FPR for the Trust Score OOD detection method evaluated on the OOD data sets for networks trained with different hyperparameters. Here ↓ implies smaller values are superior and ↑ implies larger values are superior.

Training Data Set	Epochs	LR	OOD data	AUC (%) ↑	FPR95 (%) ↓	FPR80 (%) ↓
New training set	70	0.0007	MNIST	73.4	77.0	43.3
			CorruptPOCUS	33.3	99.4	97.6
			CCA	88.6	40.4	16.7
	65	0.0007	MNIST	72.0	90.1	48.4
			CorruptPOCUS	70.9	96.4	59.3
			CCA	82.0	63.1	25.0
	50	0.0001	MNIST	26.9	100	97.7
			CorruptPOCUS	55.9	98.7	88.7
			CCA	68.9	91.7	51.2
Old training set	70	0.0007	MNIST	77.0	73.2	43.3
			CorruptPOCUS	81.5	89.8	26.4
			CCA	67.6	85.7	57.1
	50	0.0001	MNIST	57.1	99.7	96.1
			CorruptPOCUS	45.4	97.7	91.9
			CCA	56.1	100.0	81.0

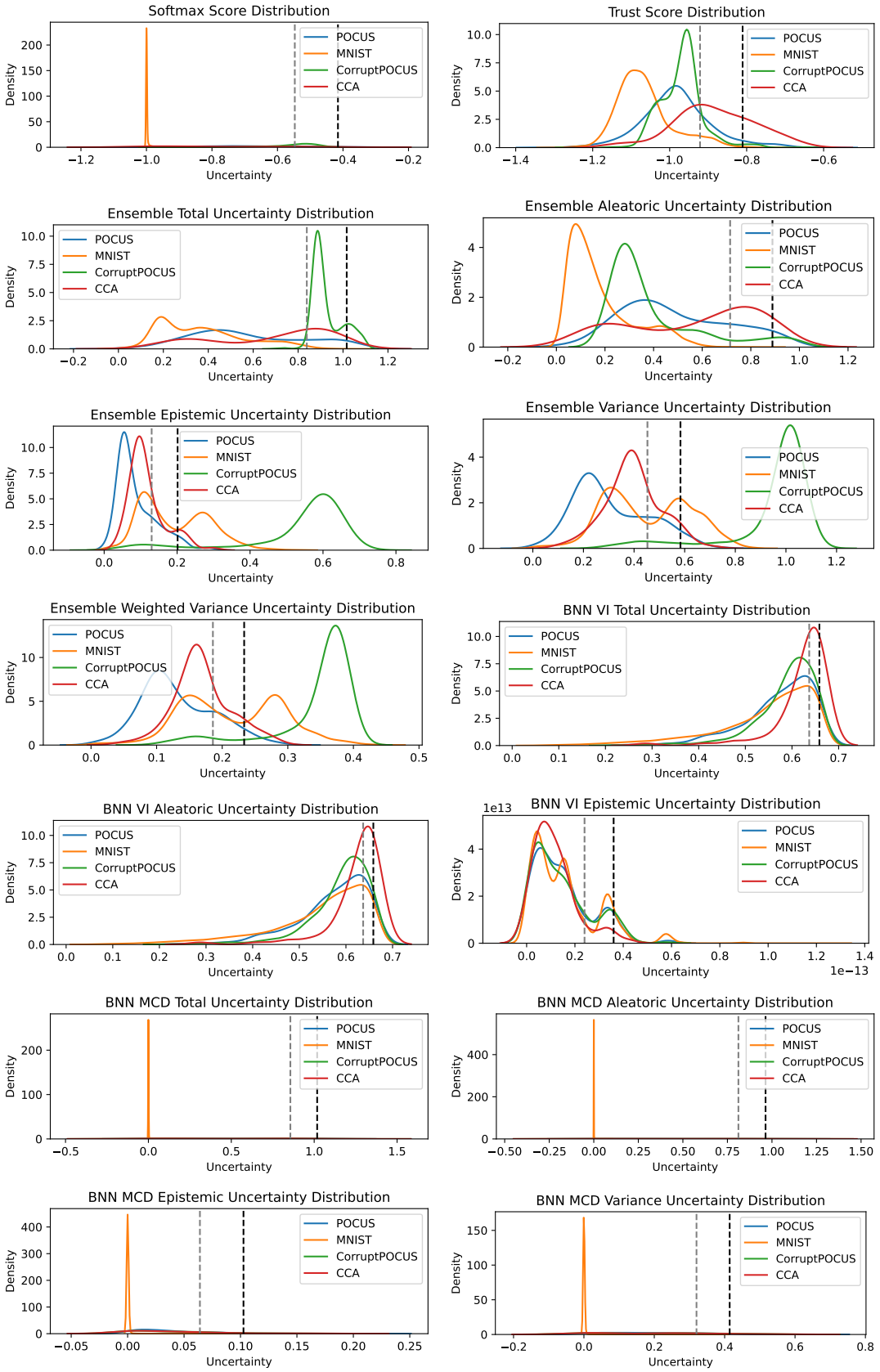

Figure 30 Uncertainty distributions for the different test sets using the different UQ methods. The vertical black line marks the threshold were 95% of the ID POCUS test data set would be classified as ID. The gray horizontal line marks the 80% respectively.

C

Energy Score Exit Comparison

Three different exits were added to the base classification network to evaluate the energies. While for the final version we used a combination of all three exits, the performances of the different exits when evaluated separately are notably divergent.

C.1 Energy Score Distributions

Each exit is trying to detect OOD samples by differentiating between energy scores from ID and OOD samples. A suitable threshold has to be found for each exit which yields a good separation between the two classes of samples. Figure 31 shows the distributions of the energy scores for each exits for the ID test set and the OOD data set. It is visible that the exits perform very differently, with exit 2 being the only one that has somewhat different energy score distributions for all data sets, i.e. the only one where a threshold can be found that will work on all data sets.

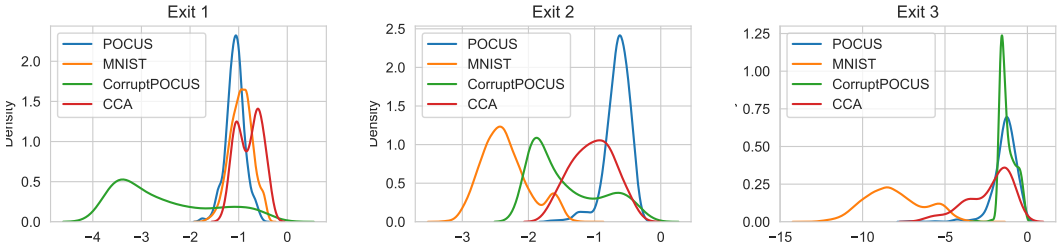

Figure 31 Distribution of energy scores from the different exits for the different data sets.

C.2 Uncertainty Quantification Experiment

In the UQ experiment, the energy score method using the combination of the three exits showed that while it might not be the best method at separating true predictions from wrong predictions, it still showed a tendency that the the energy score can be used as a measure for trustworthiness (see Tables 6 and 7). Looking at the exits separately however shows that none of the exits alone performs well at the UQ experiment (Table 15). None of the exits shows the desired correlation between the energy score and the correctness of the prediction. Specifically, the last exit shows an opposite correlation, where the results on the subset with the highest uncertainties are perfect, and the results get gradually worse for the subsets with lower uncertainties. For the first two exits, there is no pattern visible, implying that there is no correlation between the energy scores from that exit and the correctness of the prediction. This yields the question where the correct pattern in the results for the combined energy score method comes from.

Table 15 AUC and accuracy on subsets of data sorted by their energy score. The subsets represent the uncertainties, where 0-20% is the smallest uncertainties, which should correlate with the best performance. Since the energy score represents certainties instead of uncertainties, they have been flipped.

	Exit	0-20%	20-40%	40-60%	60-80%	80-100%
ACC (%) ↑	Exit 1	68.9	67.6	65.1	79.8	57.8
	Exit 2	57.1	64.7	56.9	62.4	78.8
	Exit 3	53.1	57.3	66.6	72.2	100.0
AUC (%) ↑	Exit 1	83.0	89.8	95.0	99.3	97.4
	Exit 2	94.0	95.3	92.7	94.1	93.1
	Exit 3	61.7	90.0	96.7	99.5	100.0

C.3 OOD Detection Experiment

The result of the OOD detection using the energy scores from just one exit at a time are in alignment with the what was observed in the distribution of the energies in Figure 31. The different exits perform well for different data sets. MNIST cannot be detected by the first exit, but the last two exits are perfect at separating MNIST data samples from ID data samples. The CorruptPOCUS data set is detected quite well in the first two exits, with the first one performing slightly better. The last exit fails at detecting almost all CorruptPOCUS samples as OOD. The CCA data set is the hardest to detect, with none of the exits performing really well. The first exits fails almost completely at detecting CCA samples as OOD. The second exit performs the best, but is still not very reliable. The last exit performs roughly around chance level. The AUC, FPR95 and FPR80 for the OOD detection experiment using the differnt exits are displayed in Table 16. From the results, one can conclude that the different exits find different types of OOD samples, which are OOD due to different reasons or patterns. Very obvious OOD data samples like spackle noise patterns in the CorruptPOCUS data set are detection early on in the network, but cannot be detected later in the network. OOD samples that are harder to distinguish from the ID data, like CCA, can be found only later in the network, but not in the beginning. Combining the results from the three different exits with a voting system where a sample is detected as OOD if at least one exits marks it as OOD, therefore is a well-working solution to yield good results on different types of OOD data.

Table 16 AUC and FPR for each exit and OOD data set.

OOD data	Exit	AUC (%) ↑	FPR95 (%) ↑	FPR80 (%) ↑
MNIST	Exit 1	34.1	97.5	86.8
	Exit 2	100.0	0.0	0.0
	Exit 3	99.9	0.0	0.0
CorruptPOCUS	Exit 1	87.1	20.0	15.8
	Exit 2	87.0	24.9	18.8
	Exit 3	48.2	99.8	97.9
CCA	Exit 1	20.1	100.0	97.6
	Exit 2	84.6	46.4	23.8
	Exit 3	71.8	64.3	48.8