POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering

Master's Degree Thesis

Uncertainty quantification in ultrasound image reconstruction

Supervisors

Prof. Kristen Mariko MEIBURGER

Prof. Silvia SEONI

Prof. Massimo SALVI

Candidate

Tomaso SECHI

July 2025

Abstract

Ultrasound imaging is one of the most commonly used modalities in clinical practice due to its non-invasive nature and the absence of ionizing radiation. Such images are typically produced with well-established algorithms for converting acoustic signals into visual outputs. In recent years, artificial intelligence has experienced remarkable growth in the field of medical imaging, particularly through the application of deep learning techniques. These models have shown high performance and the potential to transform the field with novel methods for image reconstruction from raw data. However, their native lack of explainability is a major limitation, and issues regarding their reliability and clinical uptake are raised. As such, there has been increasing effort in recent years to develop uncertainty quantification metrics that can complement medical decision-making by providing insight into the confidence of model predictions.

The aim of this thesis was to investigate the possibility of defining an uncertainty measure that can quantify the confidence of image reconstructions performed by a deep learning model on ultrasound data to create a measure that would enable clinicians and scholars to establish the reliability of AI-produced reconstructions. The employed database was based on a collection of 8,009 raw ultrasound data consisting of 8,000 simulated samples and 9 real ultrasound acquisitions. The reconstruction was performed based on a modified U-Net network consisting of two independent decoder branches that were designed to perform image reconstruction and anatomical segmentation simultaneously from raw ultrasound signals.

Monte Carlo Dropout was used to generate multiple stochastic reconstructions by repeating inference with dropout active. The dual-output network architecture of the network was utilized to develop an uncertainty measure that puts greater importance on the areas determined to be more relevant by the model by assigning greater value to the uncertainty in the segmented ares. This resulted in developing a new measure called Median of Weighted Uncertainty on the Reconstruction (MWUR). The uses of the measure as a loss for better reconstruction in network training were also examined.

The network obtained a mean absolute error (MAE) of 0.0693 ± 0.0130 when operating with its baseline loss function and 0.0684 ± 0.0123 when trained on the MWUR-based loss on simulated data. On real data, the corresponding MAE was 0.1189 ± 0.0289 and 0.1305 ± 0.0334. The MWUR measure proved to be effective in discriminating between inferences drawn from simulated data and actual data, and also in showing a correlation with mean absolute error.

The measure has proven to be promising to quantify the uncertainty of ultrasound images reconstructed directly from raw signals by deep learning networks, for

different configurations. However, the MWUR measure is based on segmentation results, limiting its usage to some network structures. Additionally, the two-branch network requires more extensive training time. Despite these limitations, MWUR has the potential to enhance uncertainty estimation in deep learning-based ultrasound image reconstruction and by providing a weighted measure focused on relevant regions, it offers more informative feedback on the reliability of reconstructed images. The results obtained with this metric have been promising, and future work could focus on training improvements to address its limitations and on developing a new, more flexible metric.

Acknowledgements

In questa sezione desidero ringraziare coloro che mi hanno supportato in un modo o nell'altro in questo percorso di studi.

Ringrazio i miei relatori Prof. Meiburger, Prof. Salvi e Prof. Seoni, che mi hanno guidato nella stesura di questa tesi e mi hanno sostenuto anche quando il lavoro comportava orari e momenti scomodi.

Ringrazio i miei Peperoni: Anna, Aurora, Andrea, Pinzu, Sampino e Sara, che mi hanno accolto e aiutato a vivere l'università con più leggerezza, tra un Vinum e l'altro. Ringrazio i Terraformatori: Michela, Miriam, Giamma, Peppe e Vale, che mi hanno fatto sentire in famiglia, nonostante la lontananza.

Un grazie speciale a Daniele e Simo, i miei Jo-bro, che sono stati il motivo fondamentale per cui sono arrivato a scrivere questa tesi dopo questi anni di magistrale.

A Fra, che dopo tanti anni da quei banchi è ancora al mio fianco. Al mio ingenniere Mattia, che nei primi anni è stato un pilastro. A Davide e Luca, che in questi anni sono stati gli unici ad ascoltarmi mentre mi lamentavo dei miei Knicks.

A Gabriele, che definire fratello è ormai riduttivo: non ci sono parole per descrivere il supporto che mi hai dato in questi anni. A Giulia (Susa), una persona fantastica per il mio fratello onorario e che ho il piacere di chiamare amica.

Alla mia famiglia, che da sempre mi sprona a dare il meglio e mi consola quando le cose non vanno bene: grazie per avermi permesso di essere qui oggi a scrivere questa tesi e per avermi reso la persona che sono.

Alla mia Giulia, che con la tua sola presenza riesci a farmi riprendere nei momenti più bui. Non c'è giorno in cui non mi senta fortunato di averti al mio fianco: sei la prima cosa a cui penso quando mi sveglio e l'ultima prima di addormentarmi. Ringrazierò per sempre quella notte in Santa Giulia.

A chi non c'è più: questa strada che ho intrapreso è merito tuo. Ti penso sempre e, se come spero esiste un'altra vita, saprò a quale porta bussare per prima. Grazie per avermi permesso di portare il tuo nome.

A tutti coloro che mi hanno consentito di arrivare fin qui, un ultimo, enorme grazie.

Table of Contents

List of Tables v

List of Figures vi

Acronyms ix

1 Introductory concepts 1
- 1.1 Basics on ultrasound imaging . . . 1
  - 1.1.1 Physics of Ultrasound imaging . . . 1
  - 1.1.2 Beamforming . . . 3
- 1.2 Deep learning image reconstruction . . . 5
  - 1.2.1 Brief intro to Artificial Intelligence . . . 5
  - 1.2.2 Convolutional Neural Networks . . . 6
- 1.3 Explainability and Uncertainty quantification in artificial intelligence 8
  - 1.3.1 Introduction to Explainable AI and Uncertainty Quantification 8
  - 1.3.2 Concepts of UQ . . . 8

2 State of the Art 10
- 2.1 Deep Learning for US Reconstruction . . . 10
- 2.2 Common Deep Learning Architectures and Approaches . . . 10
- 2.3 Uncertainty Quantification in US Reconstruction . . . 12
- 2.4 Research Gaps and Objectives . . . 12

3 Materials and Methods 14
- 3.1 Data . . . 14
- 3.2 Network architecture . . . 15
  - 3.2.1 Unet . . . 16
  - 3.2.2 Chosen network architecture . . . 17
- 3.3 Uncertainty estimation metric method . . . 18
  - 3.3.1 Monte Carlo dropout . . . 18
  - 3.3.2 Uncertainty metric extraction method . . . 20

3.4 Network training and evaluation . . . 23
3.4.1 Training parameters . . . 23
3.4.2 Loss functions . . . 23
3.4.3 Network evaluation metrics . . . 24
3.5 Uncertainty metric evaluation criterion . . . 24
3.6 Empirical Analysis for Metric Selection and Model Tuning . . . 25
3.6.1 Best Drop out values for Monte Carlo dropout and best uncertainty metric selection . . . 25
3.6.2 Uncertainty metric normalization attempts . . . 25
3.6.3 Uncertainty metric as loss function during training . . . 26
3.6.4 Training and evaluation on reduced dataset . . . 28
3.7 Additional test of uncertainty metric on photoacustic images . . . 29

4 Results 31
4.1 Network Performances . . . 31
4.2 Results of uncertainty metrics for reconstruction . . . 32
4.3 Selection of drop out values and selection of best uncertainty metric 33
4.4 Results of the normalization attempts . . . 36
4.5 Performances of the network trained using MWUR as a loss . . . 37
4.6 Performances with reduced datasets . . . 38
4.6.1 network performances . . . 38
4.6.2 MWUR perfromances . . . 40
4.7 MWUR performance on the photoacustic acquisitions dataset . . . 44

5 Conclusions 45
5.1 Network Performance . . . 45
5.2 Uncertainty Metrics for Reconstruction . . . 45
5.3 Dropout Probability and Best Metric Selection . . . 46
5.4 Normalization of MWUR . . . 46
5.5 Training with MWUR as a Loss Function . . . 47
5.6 Training with Reduced Datasets . . . 48
5.7 MWUR on Reduced Datasets . . . 48
5.8 MWUR on the PA dataset . . . 48
5.9 Final Remarks . . . 48

Bibliography 50

List of Tables

3.1 Training parameters. 23
3.2 Training, validation and test set composition of each subset 28

4.1 Performance of the network on each dataset. 31
4.2 Pearson correlation and p-values of each metric in Validation, Test and Real sets. 32
4.3 Dice results comparison between the network trained with different losses 37
4.4 MAE results comparison between the network trained with different losses 37
4.5 Comparison of MAE performances on the reconstruction inferences between network MAE loss trained (**a**) and Weighted loss trained (**b**) with reduced datasets. 38
4.6 Comparison of Dice performances on the segmentation inferences between network MAE loss trained (**a**) and Weighted loss trained (**b**) with reduced datasets. 39
4.7 Comparison of MAE performances on reconstruction inferences between networks trained with MAE loss (**a**) and Weighted loss (**b**) on reduced datasets. 40
4.8 Comparison of Dice coefficient performances on reconstruction inferences between networks trained with MAE loss (**a**) and Weighted loss (**b**) on reduced datasets. 40

List of Figures

1.1 Scheme of delay and sum algorithm 3
1.2 Visual representation of difference between the different types of US image display methods 5
1.3 Relation between AI, ML and DL. 6
1.4 A generic Convolutional Neural Network to perform classification. . 6

3.1 First five raw data for validation set **(a)** 15
3.2 Original Unet architecture created by Ronneberger et al[24]. 16
3.3 Unet employed in this thesis work to perform simultaneous reconstruction and segmentation from raw US acquisitions 17
3.4 Monte Carlo dropout 18
3.5 Visual scheme of the described segmentation pipeline. **Note:** The colors used in this diagram are intended solely for explanatory purposes. . . . 21
3.6 Visual scheme of the described Reconstruction pipeline. **Note:** The colors used in this diagram are intended solely for explanatory purposes. . . . 22
3.7 Visual scheme of the first normalization attempt. 26
3.8 Schemes of the losses combination with MWUR metric 27

4.1 On the **left** an example of ground truths and outputs obtained from a simulated acquisition. On the **right** from a real acquisition. . . . 32
4.2 In this image, the results of the uncertainty metrics on the terms described in Section 3.5 are shown. In the top part the box plots are shown. In the bottom the correlation graphs between the metric and the MAE. 33
4.3 Boxplots of each metric at different values of dropout probability. . 34
4.4 Pearson correlation of each metric at different values of dropout probability. 35
4.5 MWUR metric normalization by the maximum of WUR matrix . . 36
4.6 MWUR metric normalization by the maximum of MWUR obtained on the construction set 36

4.7 Some outputs obtained on the reduced dataset using different losses during training. 39
4.8 Box plots of MWUR metric obtained using reduced datasets. At the Top training with MAE loss, at the Bottom with the Weighted loss. 41
4.9 Pearson correlation plots of MWUR metric obtained using reduced datasets. At the Top training with MAE loss, at the Bottom with the Weighted loss. 42
4.10 Box plots of MWUR metric obtained using reduced datasets with full test. At the Top training with MAE loss, at the Bottom with the Weighted loss. 42
4.11 Pearson correlation plots of MWUR metric obtained using reduced datasets with full test. At the Top training with MAE loss, at the Bottom with the Weighted loss. 43
4.12 Pearson correlation of MWUR against MAE for the PA validation set e test set. 44

Acronyms

US
Ultra Sound

EM
Electromagnetic

DAS
Delay And Sum

RF
Radio Frequency

AI
Artificial Intelligence

ML
Machine Learning

DL
Deep Learning

DNN
Deep Neural Network

ANN
Artificial Neural Network

CNN
Convoultional Neural Network

XAI

Explainability in Artificial Intelligence

UQ

Uncertainty Quantification

PU

Predictive Uncertainty

AU

Aleatoric Uncertainty

EU

Epistemic Uncertainty

MC

Monte Carlo

BNN

Bayesian Neural Network

FCN

Fully Convolutional Network

GAN

Generative Adversarial Network

WUR

Weighted Uncertainty on Reconstruction

MWUR

Median of Weighted Uncertainty on Reconstruction

MAE

Mean Absolute Error

PAI

Photoacoustic imaging

Chapter 1

Introductory concepts

1.1 Basics on ultrasound imaging

1.1.1 Physics of Ultrasound imaging

Ultrasounds (US) imaging or sonography is a medical imaging technique that has consolidated itself as a useful tool in clinical practice thanks to its ease of use and safety. US imaging is based on the interaction of an US wave generated via a probe with human tissues. Both the wave generation and detection is obtained thanks to the piezoelectric elements on the probe via piezoelctic phenomenon. This phenomenon is exhibited by some crystalline materials, and involves the reversible conversion of two forms of energy from one to the other, namely mechanical and electrical energies, so they can be used as transducers. In USs:

- The generation of the US wave is obtained by applying a potential difference to the elements using the Inverse piezoelectric effect. The elements respond to this stimuli by expanding and contracting, converting the electrical energy to mechanical energy, generating the US wave.
- The detection is based on the direct piezoelectric effect. After interacting with human tissues, the wave returns to the probe where the piezoelectric crystals convert the mechanical energy to an electrical energy.

The US that are used in medicine have center frequency f_0 in the range of 2 to 15MHz and the speed of sound c is around 1540 m/s.[1] The wavelength λ can be obtained by the relation:

$$\lambda = \frac{c}{f_0} \tag{1.1}$$

λ is also a measure of the maximum spatial resolution. In fact, structures with dimensions smaller than the wavelength can not interact with the US and, thus,

can't be detectable. Depending on how much their medium particles withstand change as a result of mechanical disturbance, different materials react differently to ultrasonic probing. This property of the medium is called characteristic acoustic impedance Z and can be defined as product of medium density and US velocity in the medium.

$$Z = density \times velocity \tag{1.2}$$

The changes of Z across the tissues are of extreme importance in US interactions. The points where such changes occur are called acoustic boundaries or tissue interfaces. When a beam of US strikes a tissue interface, part of the beam energy is transmitted across the interface, while some is redirected backwards, or reflected. The reflected beam is referred as echo and its intensity is determined by the angle of incidence and the difference in acoustic impedance between the two mediums at the boundary. The difference in Z value is also known as the acoustic mismatch. When the beam strikes a reflector perpendicular to the surface of the acoustic boundary, the angle of reflection is equal to 0 and the echo goes straight back to the transducer. In this special case the echo intensity in relation to the intensity of the US beam incident upon the boundary is given by the relation:

$$\frac{I_r}{I_i} = \frac{(Z_1 - Z_2)^2}{(Z_1 + Z_2)^2} \tag{1.3}$$

- I_r = intensity of reflected echo.
- I_i = intensity of incident beam at the boundary.
- $Z_{1,2}$ = acoustic impedance of first and second medium

The ratio Ir/li is known as the reflection coefficient. It represents the proportion of beam intensity which is reflected from the interface. A substantial change in Z value at an interface increases the reflection coefficient, resulting in a more intense echo, whereas minor variations in Z produce smaller echoes. This inference is crucial in ultrasonic imaging[2]. The transducer's US beam can either be a plane wave or a focused wave. If it is a focused wave, the US beam is steered across the subject plane by activating small groups of crystal elements in a sequential manner. Using a pulsing technique with precisely controlled time delays between the various elements, the US waves radiating from each of the crystals in the array can be caused to arrive in phase at a specific focal point. At the focused point, the waves interfere constructively and produce the formation of a region of high intensity. The ultimate image is formed by the scanning of discrete lines in alignment. Aside from any possible delays with the reconstruction or enhancement of images, the technique allows one to record a single frame after sequential transmission of beams and receiving backscattered echoes by each transducer channel. The primary drawback

of the technique is the low frame rate: as image acquisition time is prolonged, the likelihood of motion artifacts also rises. In plane-wave imaging, no sequential scanning of the medium line by line is involved; rather, it uses multiple channels transmitting in parallel with no delay. Hence, the whole imaging area can be acquired within the time it takes to scan a single line. As plane-wave imaging utilizes all the transducer elements for simultaneous transmission, it enables us to acquire US data at frame rates of up to several thousand frames per second. Yet, this method may worsen image quality by the lack of transmission focusing.[3]

1.1.2 Beamforming in US imaging

Beamforming is a signal processing technique that enables the steering, shaping, and focusing of an electromagnetic (EM) wave using a sensor array, which points the wave in the direction of a specific target.[4]. In US medical imaging, beamforming is primarily involved with shaping the spatial distribution of the amplitude of the pressure field in the area of interest and subsequently recombining the reflected US signals to create images.[5].

DELAY-AND-SUM

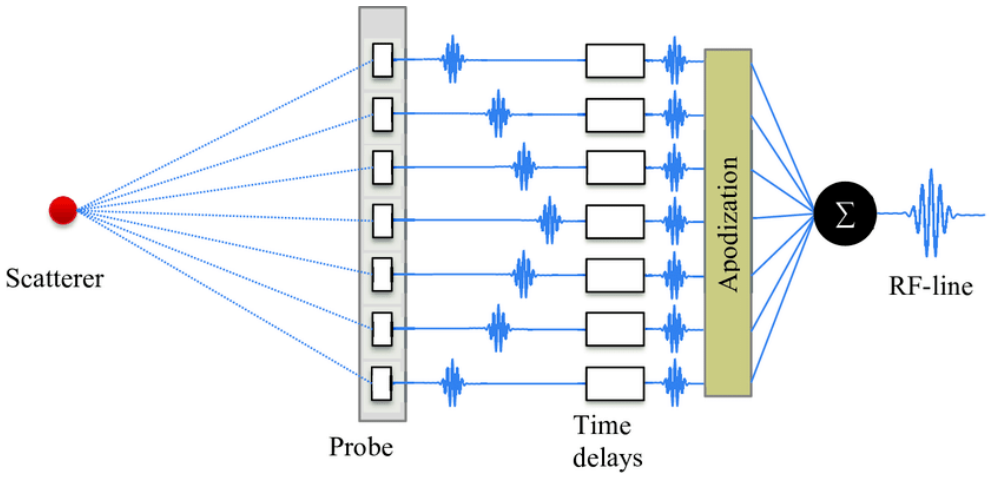


Figure 1.1: Scheme of delay and sum algorithm

DELAY-AND-SUM (DAS) is the most basic digital beamformer for medical US imaging. Because of its simplicity and efficiency, it's the most used algorithm to beamform the us signal. The idea of DAS is to collect and reunite all the Radio Frequency (RF) signals generated from a single scatterer. The x-axis follows the same direction of the transducers array, while the z-axis represents the depth direction. Let's define RF (x_1, t) as the signal sensed by the receiving transducer

in position x_1 on the x-axis at time t. Let's also separate the propagation in the medium in two stages: the travel from the transducer to the scatterer and the travel from the scatterer back to the transducer. Finally let's define a single scatterer in the medium at the position (x, z). If no steering angle is applied to the plane wave, the times τ required for the two listed stages are the following:

- **Trasmission:**

$$\tau_1(x, z) = \frac{z}{c} \tag{1.4}$$

- **Reception:**

$$\tau_2(x_1, x, z) = \frac{\sqrt{z^2 + (x - x_1)^2}}{c} \tag{1.5}$$

Where c is the speed in the tissues that is assumed constant and equal to 1540 m/s. Hence, the total time required for the wave to propagate in the medium and to go back to the transducer is:

$$\tau = \tau_1 + \tau_2 \tag{1.6}$$

Finally, in order to reconstruct the scatter point in the position (xz), it is only needed to delay the echoes by τ and coherently sum them[6]. One the information has been processed it can be displayed by using different methods denominated modes. The most commonly used modes are:

- **A-mode:** the signals from returning echoes are displayed in the form of spikes on a cathode ray oscilloscope (CRO), traced along a time base. While the Amplitude of the spike is a measure of the echo size, its position is a measure of the distance of the transducer form the related reflecting boundary. This mode display only 1-D information, so it does not produce an image.

- **B-mode:** This is a mode most associated to the term US image. The returning signals from a single scan-line are displayed as pixels of diverse intensities. Each pixel is intensity (brightness) is a measure of the echo size, while the position is a measure of the distance of the reflector from the transducer. The combined information of multiple scan-lines provides a 2-D image of the cross-section of the target.

- **M-mode:** Used to generate a trace of a moving reflector along the path of the US beam.

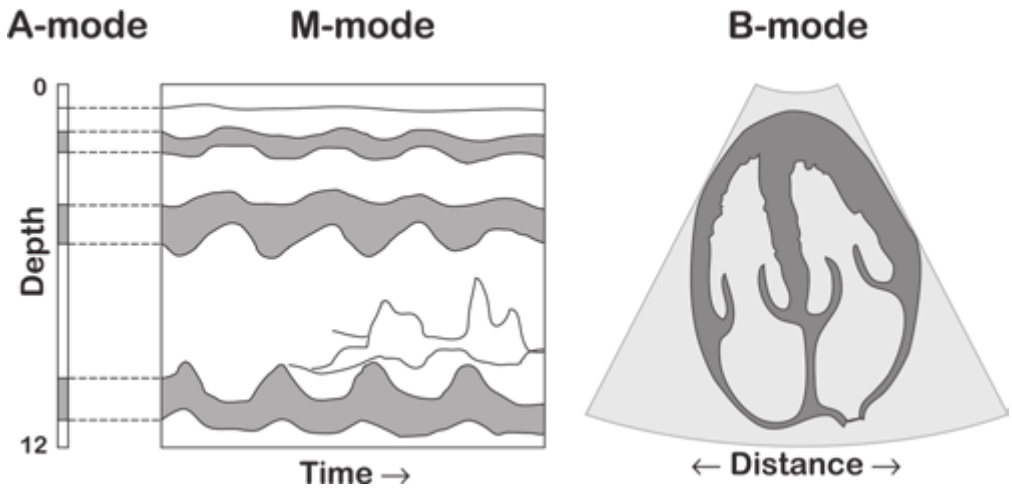


Figure 1.2: Visual representation of difference between the different types of US image display methods

1.2 Deep learning image reconstruction

1.2.1 Brief intro to Artificial Intelligence

The field of Artificial Intelligence (AI) is a field of computer science that focuses on the creation of systems capable of performing tasks that usually require human mental capabilities. AI is an area that contains a broad range of computational approaches, including rule-based systems, optimization methods, and learning-based approaches. AI is used as an overarching term that includes both Machine Learning (ML) and Deep Learning (DL).

ML is a branch of AI that allows machines to learn from data without being explicitly programmed. It consists of algorithms that enhance performance incrementally by reducing mistakes and increasing predictive precision. ML systems are dynamic in adaptation, with their decision-making processes continually being adjusted as they acquire more data. A typical pipeline of building an ML model consist of three phases: training, validation and testing. During the training phase a set of the data called training set is given as an input to the model, who will learn the parameters to efficiently perform a given task. In the validation a second set called validation set, fine-tune will be performed based on this set. Finally in the test phase the objective and unbiased performance of the model on an independent data set (ie,the testing data set) are evaluated. DL, a distinct branch of Machine Learning (ML), employs artificial neural networks (ANNs) that are inspired by the composition and functionality of the human brain. It consists of several layers of interconnected nodes—i.e., input, hidden, and output layers—to learn patterns

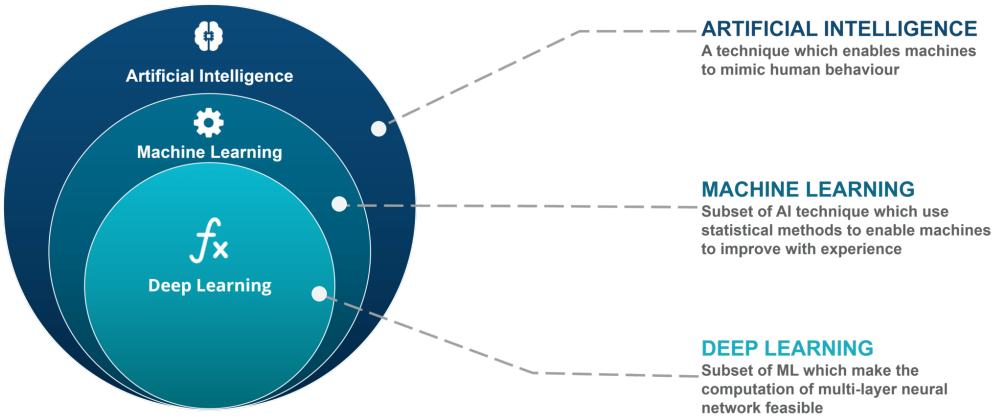


Figure 1.3: Relation between AI, ML and DL.

and hierarchical features from raw data. "Deep" implies the employment of more than one hidden layer, which allows the system to learn intricate representations. While DL excels at processing unstructured data with high precision, it demands lots of computing power and enormous amounts of training data [7].

1.2.2 Convolutional Neural Networks

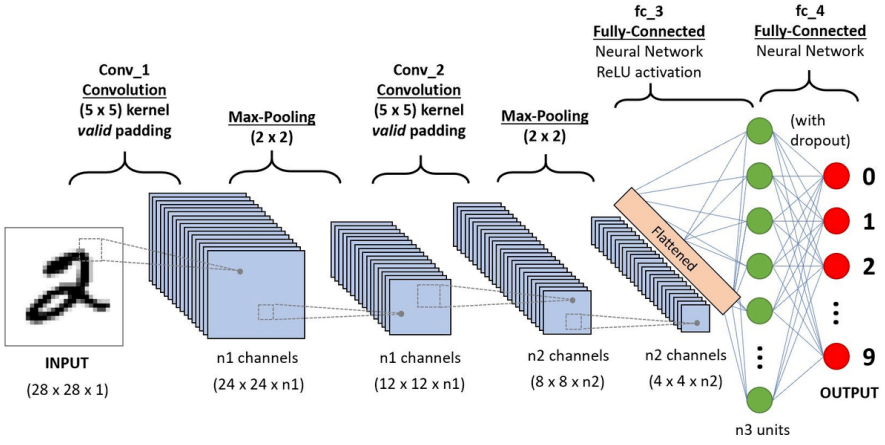


Figure 1.4: A generic Convolutional Neural Network to perform classification.

The ANNs are systems inspired by how the human nervous system operates. They are comprised of multiple interconnected computational nodes called neurons that work in junction to learn from input data to optimize the network output. In the basic structure of a ANN, the input is loaded to the input layer l of which

will distribute it to the hidden layers. Then, these will make decisions from the previous layer and weigh up how a stochastic change within itself detriments or improves the final output. Convolutional Neural Networks (CNN) are analogous to traditional ANNs in that they are comprised of neurons that self-optimize through learning. The most notable difference of CNNs to the ANNs is that the firsts are mostly used in the image pattern recognition field. The neurons that CNN are comprised of, are organized into three dimensions, the spatial dimensionality of the input (height and the width) and the depth. CNNs are comprised of three types of layers[8]. These are:

- **convolutional layers:** The name CNN comes from these particular kind of layers of which they are comprised. The layers parameters focus around the use of learnable kernels. These kernels have a limited spatial dimensions yet cover the complete depth of the input. When data enters a convolutional layer, it convolves each filter across the input's spatial dimensions to generate a 2D activation map. As we progress through the input, the scalar product is computed for each value in that kernel. The network will then learn kernels that 'fire' when they detect a specific characteristic at a given spatial position in the input. These are frequently referred to as activations. Each kernel will have an activation map, which will be layered along the depth dimension to generate the convolutional layer's total output volume.
- **pooling layers:** Pooling layers attempt to gradually lower the dimensionality of the representation, hence reducing the number of parameters and computing complexity of the model. The pooling layer operates over each activation map in the input, scaling its dimensionality according to a function (commonly the MAX or Average).
- **fully-connected layers:** The fully-connected layer contains neurons that are directly connected to the neurons in the two adjacent layers, but not to any layers inside them. This is similar to how neurons are placed in conventional kinds of ANN.

The CNN found application in a wide range of task applied to images. The most common task in the clinical field of which they are applied are:

- **Classification:** Assignment of a label to the whole image;
- **Regression:** Similar to classification, but outputs a continuous real number.
- **Detection:** The detection task identifies a target object by outputting a bounding box enclosing it;
- **Segmentation:** could beconsidered as a dense classification in which each pixel is classified into a label.

- **Reconstruction:** is the process of creating a target domain image from source domain signals. It may be either raw-to-image or postprocessing. The first method creates an image based on raw sensor data, whereas the second method takes advantage of extracted features of source domain images [9].

1.3 Explainability and Uncertainty quantification in artificial intelligence

1.3.1 Introduction to Explainable AI and Uncertainty Quantification

The inherent complexity and "black box" nature of DL has risen concern about its trustworthiness and reliability, especially in the medical field. A model for instance could learn incorrect or low relevant feature and rely on them during its prediction instead of the clinically relevant ones. This scenario open the need of finding way to understand or/and quantify the quality of the decision making process of the DL network. Explainable Artificial Intelligence (XAI) is a broad term that encompasses the methods that aim to make the AI decision more transparent to human users by providing additional data beyond the network output to understand how the model has arrived to the decision. Uncertainty Quantification (UQ) instead tries to quantitatively measure the uncertainty associated with the model prediction to asses the model reliability, so is the process of determining the extent to which model's predictions may be uncertain or unreliable [10].

1.3.2 Concepts of UQ

As stated in the previous section, is crucial to evaluate the uncertainty of AI system predictions. The concept of uncertainty is bound to the grade of ambiguity or confidence on the outputs of AI models. This uncertainty can be the caused by multiple factors such as noisy training data sources, limited domain knowledge or by the model itself. Predictive uncertainty (PU) is used to determine this uncertainty and can be divided:

- **Aleatoric Uncertainty (AU):** Stemming from inherent data noise or randomness. This not a property of the model, but rather is an inherent property of the data distribution, and hence, it is irreducible.
- **Epistemic Uncertainty (EU):** Arising from limited knowledge or data scarcity, leading to uncertainty about the model's behavior or performance in new or unseen situations. This uncertainty can be reduced by improving architecture, learning process or data quality.

[11] So the PU can be represented as the sum of these two uncertainty types

$$PU = AU + EU[UQ] \tag{1.7}$$

To estimate uncertainties implicitly embedded in models, Bayesian inference provides an immediate remedy and stands out as the main approach. Bayesian methods have risen to interest due to their characteristic of beaing able to address uncertainty via posterior distribution, reduce overfitting and for enabling sequential learning while retaining prior and past knowledge. The most challenging task in following the Bayesian paradigm is the computation of the posterior. In the typical ML setting characterized by a high number of parameters and a considerable size of data The Bayesian paradigm is based on two simple ideas. The first is that probability is a measure of belief in the occurrence of events, rather than just some limit in the frequency of occurrence when the number of samples goes toward infinity. The second is that prior beliefs influence posterior beliefs. The above two are summarized in the Bayes theorem: Let D denote the data and $p(D|\theta)$ the likelihood of the data based on a postulated model with $\theta \in \Theta$ a k-dimensional vector of model parameters. Let $p(\theta)$ be the prior distribution on θ. The posterior distribution $p(\theta|D)$:

$$p(\theta|D) = \frac{p(D|\theta)}{p(D)} = \frac{p(\theta)p(D|\theta)}{p(D} \tag{1.8}$$

where $p(D) = \int_\Theta p(\theta)p(D|\theta)\, d\theta$ is called marginal likelihood and is a normalization constant. With the Bayesian approach the unknown variable θ is treated as a random variable. The prior probability $p(\theta)$, which intuitively expresses in probabilistic terms any knowledge about the parameter before the data has been collected, is updated in the posterior probability $p(\theta|D)$, mixturing prior knowledge and evidence supported by the data through the model's likelihood. Bayesian inference is generally difficult due to the fact that the marginal likelihood is often intractable and of unknown form, resulting in difficulty in computing an exact posterior inference, but it can be approximated with methods such as Monte Carlo (MC) dropout [12].

Chapter 2

State of the Art

2.1 Deep Learning for US Reconstruction

US imaging is a widely used modality in clinical diagnostics, appreciated for its non-ionizing nature, portability, and relatively low cost. Conventional US image formation relies heavily on DAS beamforming techniques, which are simple and fast but often lead to limited spatial resolution, low contrast, and high levels of speckle noise and imaging artifacts[13, 14].

In recent years, DL methods have revolutionized image processing and analysis in medical imaging. The application of DL to US imaging has sparked significant interest, primarily due to its ability to learn complex nonlinear mappings directly from data, enabling the generation of higher quality images compared to traditional signal processing approaches[15, 16].

However, a notable ambiguity exists in the literature regarding the terms *image enhancement* and *image reconstruction*. Enhancement typically refers to post-processing methods that improve image quality from already beamformed data, such as DAS outputs, through denoising, contrast improvement, or speckle reduction[17]. Reconstruction, in contrast, implies a more fundamental transformation where the neural network learns to directly map raw RF or channel data into B-mode images, potentially replacing or modifying the beamforming process itself[18, 19].

2.2 Common Deep Learning Architectures and Approaches

The most common DL architectures employed for US image reconstruction are:

- **U-Net and Fully Convolutional Networks (FCNs)**: These are typically used for end-to-end mapping from raw data to image space. U-Nets with

encoder-decoder structures and skip connections have been successfully applied to suppress speckle noise and improve structural preservation[20, 17].

A particularly promising direction involves the use of multitask U-Net architectures that simultaneously perform image reconstruction and semantic segmentation. For instance, Rivaz et al.[19] proposed a model that takes raw channel data as input and outputs both a reconstructed B-mode image and a corresponding segmentation map. This multitask learning paradigm exploits shared representations to improve both tasks: segmentation enhances reconstruction by enforcing anatomical consistency, while improved reconstructions benefit segmentation through higher signal clarity.

Similarly, Goudarzi and Rivaz[16] demonstrated that networks trained jointly on simulated and in vivo data can achieve generalizable performance in both domains, further underscoring the clinical viability of this approach. These models often leverage U-Net backbones with task-specific output branches and are optimized using composite loss functions combining pixel-wise reconstruction and segmentation objectives.

- **Autoencoders**: Convolutional autoencoders (CAEs) have been trained to learn the residual between noisy DAS images and high-quality minimum variance (MV) outputs, effectively denoising and sharpening image outputs[13].

- **Generative Adversarial Networks (GANs)**: GANs have been used to generate realistic B-mode images by leveraging adversarial loss, improving the perceptual quality and generalization to real or phantom data[21].

- **Frequency-domain networks**: Some works apply DNNs to frequency-domain representations, allowing better suppression of off-axis and reverberation artifacts. Luchies et al. demonstrated improved contrast resolution using this approach[14].

- **Compressed and Sub-Nyquist Reconstruction**: Networks such as the one proposed by Mamistvalov et al. recover high-quality B-mode images from spatially and temporally sub-sampled RF data, addressing both hardware and computational constraints[18].

- **Multitask Learning**: Dahan et al. propose a multitask CNN that performs both beamforming and denoising, adjusting its weight normalization to accommodate different image quality tasks without sacrificing frame rate[17].
  “‘

2.3 Uncertainty Quantification in US Reconstruction

UQ is an emerging yet underdeveloped aspect in deep learning-based medical US. As these models move toward clinical deployment, their reliability and interpretability become crucial. UQ methods aim to quantify the confidence of neural networks in their predictions, particularly valuable when models are applied to out-of-distribution data or ambiguous regions[22].

In the context of US, Haji-Saeed et al. proposed a Bayesian DL framework for passive cavitation imaging using MC dropout, which estimates both epistemic and aleatoric uncertainty[22]. Their results showed that uncertainty maps correlate with regions prone to artifacts, making them useful for both clinical interpretation and automated downstream tasks such as segmentation.

Standard UQ methods used in medical imaging include:

- **Monte Carlo Dropout**: Applying dropout at inference to approximate Bayesian inference.
- **Deep Ensembles**: Using multiple independently trained models to estimate uncertainty.
- **Test-Time Augmentation**: Assessing prediction variability under input transformations.
- **BNNs**: Modeling weights as distributions rather than fixed values.

Despite its importance, UQ is still rarely integrated in US image reconstruction pipelines. Incorporating uncertainty estimates could provide a crucial reliability layer in AI-driven clinical systems.

2.4 Research Gaps and Objectives

The reviewed literature clearly demonstrates the potential of DL for US image reconstruction. However, important gaps remain:

- The distinction between enhancement and reconstruction is not consistently defined, leading to methodological confusion.
- Most models are trained on simulated datasets with limited diversity, raising concerns about domain generalization.
- Few studies incorporate uncertainty estimation in reconstruction models, despite its clinical significance.

This thesis aims to address these gaps by proposing a DL-based reconstruction pipeline that directly reconstructs US images from raw RF data, while integrating principled uncertainty quantification to enhance model transparency and robustness.

Chapter 3

Materials and Methods

3.1 Data

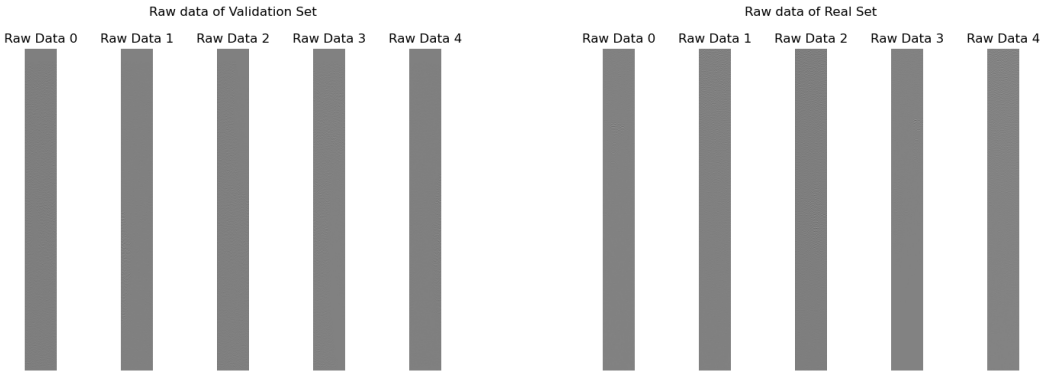
The dataset consisted of 8000 simulated US acquisitions and 9 real acquisitions. The *Field II US simulation package* [23] was used to generate individual anechoic cysts surrounded by homogeneous tissue. The use of simulations in the application of deep learning for US is common for two primary reasons:

- this enables the generation of large, diverse data sets that are required to train robust DNNs.
- in segmentation tasks, simulations enable the specification of ground-truth pixel labels, allowing one to avoid the expensive and time-consuming step of a human annotator to provide segmentation labels.

The real set is considerably smaller than the other, but is still a very important set as it gives the opportunity to measure differences in performances obtained in a real scenario. It is composed by US acquisition made on a phantom and of charotid-zone acquisitions on human subjects. These acquisition are crucial to evaluate the performances of the uncertainty metrics that were constructed during the curse of this thesis work. In fact, as the real scenarios are more complex and noisy than the simulated ones, this should be reflected in a growth in the uncertainty of the results. The simulated data was randomly splitted accordingly to this table:

Set name	Percentage of sim. dataset	Set dimension
Training set	50%	6400
Validation set	25%	800
Test set	25%	800

While the real data was used to construct a second test set and was limited to the test phase. Each singular raw acquisition is two dimensional, so can be visualized as an image of dimensions 1290x128. For each simulated acquisition there are two target images, one for the reconstructed image (called target or prediction) and one for its own segmentation (called segmentation target); these two targets are the ground truth for this set. For real acquisitions, the targets of the predictions are obtained with the DAS algorithm and no segmentation targets where used. All the target images for both the segmentation and the reconstruction are 2-D images of dimension 256x128 pixels.



(a) Validation data **(b)** Real data

Figure 3.1: First five raw data for validation set **(a)** and the Real set **(b)**

3.2 Network architecture

As seen in the state of the art chapter, for the image reconstruction task the most commonly used approaches in literature are:

- GAN
- Pre produce DAS image and than pass it to the network
- Use raw data as input to a network to obtain its reconstruction

While the GANs are a valid choice, the final objective of this thesis is to obtain a uncertainty metric and UQ in this kind of architecture are more complex, so this approach was not employed. The second approch is more akin to a task of image enhancement that in literature is often confused with the reconstruction, as the objective was to evaluate uncertainty on the reconstructed images directly from raw data, the only valid approach was the last one in the list.

3.2.1 Unet

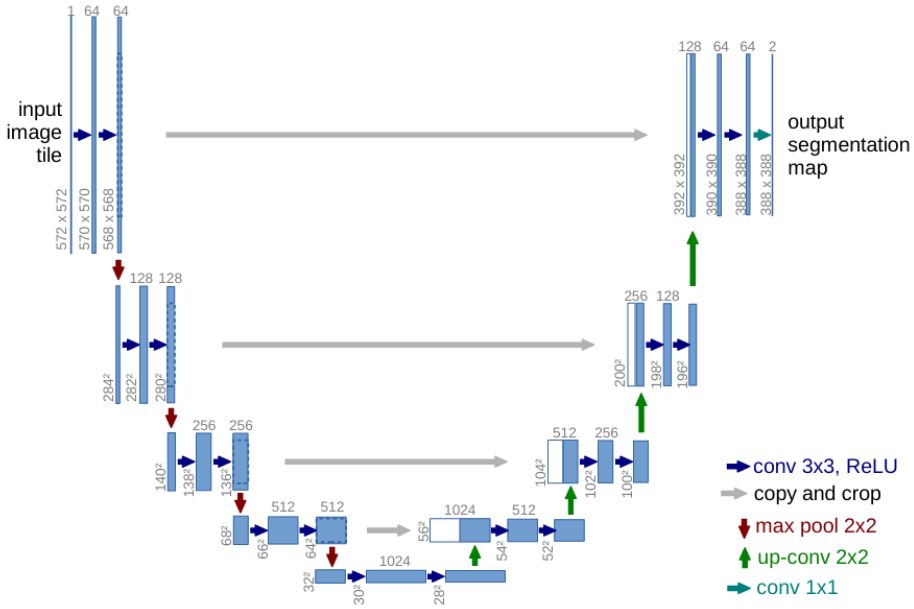


Figure 3.2: Original Unet architecture created by Ronneberger et al[24].

Almost the totality of similar works used some variation of the Unet to reconstruct images. The Unet is one of the most commonly used CNN in medical imaging applications. The network architecture consists of a contracting path and an expansive path. The main feature of the Unet are the skip connection between the two paths that are used to combine the high level feature of the contracting path with the low level features of the expanding path. This connection of feature makes the network particularly suitable to medical applications. The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (un-padded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution (up-convolution) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64 component feature vector to the desired number of classes. In total the network has 23 convolutional layers [24].

3.2.2 Chosen network architecture

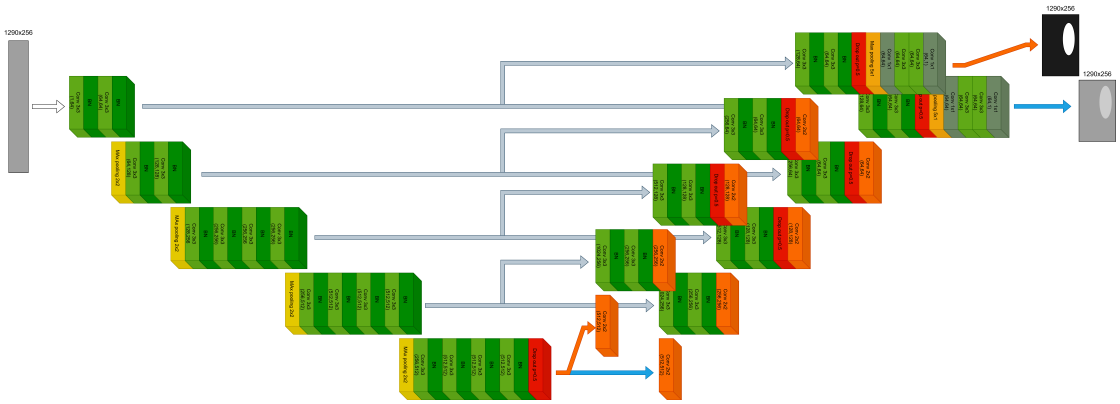


Figure 3.3: Unet employed in this thesis work to perform simultaneous reconstruction and segmentation from raw US acquisitions

The network used on this thesis work is a variation of the Unet, inspired by the works of [25]. The implemented deep learning architecture is a dual-branch encoder-decoder convolutional neural network designed to simultaneously perform image reconstruction and semantic segmentation. The network follows a U-Net-like structure, featuring a shared encoder path and two separate expansive decoder branches.

The encoder (contraction path) is composed of five stages of convolutional blocks with ReLU activations, batch normalization, and max pooling operations. Each stage increases the number of feature maps while reducing the spatial resolution, enabling the model to capture hierarchical features. To improve feature extraction, deeper blocks are introduced at lower resolutions. Dropout regularization is applied at the bottleneck to reduce overfitting.

The first decoder branch is tailored for image reconstruction. It leverages transposed convolution layers (deconvolutions) for upsampling, followed by concatenation with corresponding encoder feature maps (skip connections). This structure allows for the recovery of fine-grained spatial details lost during encoding. After each concatenation, standard convolutional blocks refine the upsampled feature maps. The reconstructed output is passed through a sigmoid-activated final convolutional layer to produce the enhanced image.

The second decoder branch is used for semantic segmentation, mirroring the reconstruction decoder in structure. It independently processes the shared encoder features through its own upsampling, convolution, and normalization layers. The segmentation output is also generated via a final sigmoid-activated convolutional layer, yielding a pixel-wise binary segmentation map

Dropout layers are placed at the end of the contracting path and after each

of the convolutional blocks of both of the two decoder branches. These layers discourages the network from becoming overly reliant on specific neurons, thereby enhancing its ability to generalize, thus avoiding overfitting. Moreover these layers are crucial to perform Montecarlo dropout and to construct the uncertainty metric.

The specific constitution of each of the block can be seen in Figure 3.3.

3.3 Uncertainty estimation metric method

The absence of relevant works on the estimation of the uncertainty in the reconstruction of US images from raw RF data provided substantial freedom in the methodological approach, but also posed challenges due to the lack of established references or guidelines to build upon. For the motives above, MC dropout, the most commonly used uncertainty estimation method, was used as a basis to extract uncertainty, then, by exploiting the dual output of the network, several experiments were exploited to define multiple uncertainty metrics.

3.3.1 Monte Carlo dropout

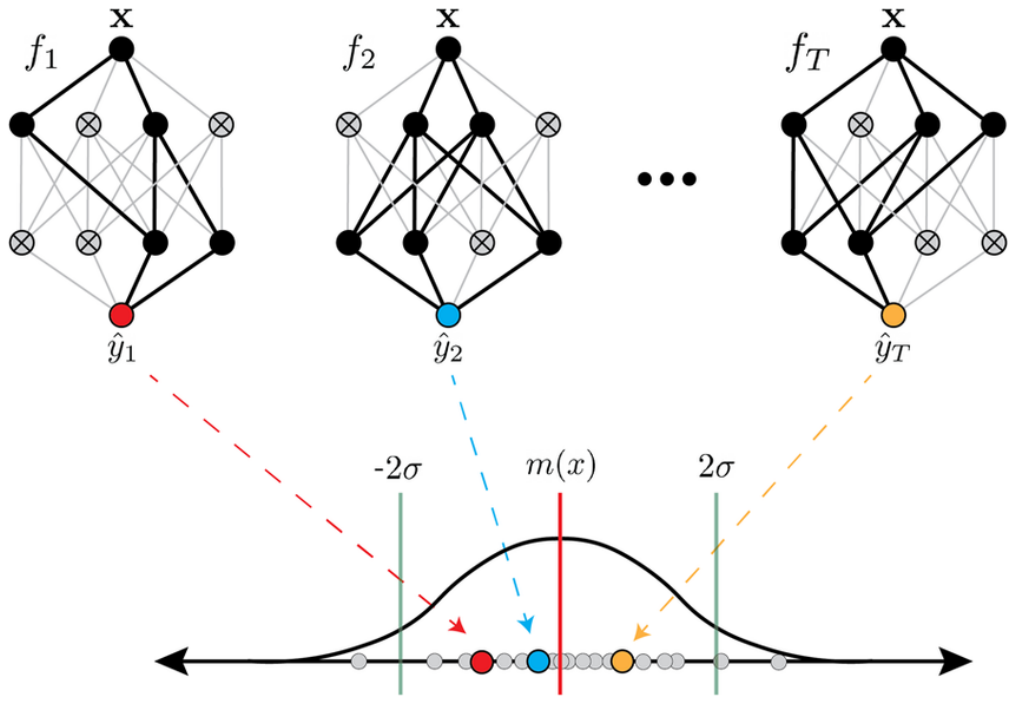


Figure 3.4: Monte Carlo dropout

To estimate the predictive uncertainty of the model, we adopted the approach

proposed by Yarin Gal and Zoubin Ghahramani, known as Monte Carlo dropout. In their work, [26] Monte Carlo dropout showed that a neural network with arbitrary depth and non-linearities, with dropout applied before every weight layer, is mathematically equivalent to an approximation to the probabilistic deep Gaussian process. Dropout is a technique commonly used in neural networks during the training phase to avoid or mitigate overfitting resulting from the co-adaptation of adjacent neurons. Practically speaking applying drop out means that, during training, neurons at each input have a random probability to be momentarily turned off to create a "thinned" version of the net consisting of the surviving units, so for a neural net with n units, 2^n possible thinned neural networks are obtained. Training a network with dropout and using this approximate averaging method at test time leads to significantly lower generalization error [27]. With MC dropout the information deriving from applying dropout that is normally discarded, can be used to derive uncertainty at test time.

Mathematical Formulation of MC Dropout

In this paragraph will be described the core mathematical basis of the MC dropout method. Assume a NN with L layers, which W_l, b_l and K_l denote the weight matrices, bias vectors and dimensions of the l_{th} layer, respectively. The output of NN and target class of the i_{th} input $x_i (i = 1, ..., N)$ are indicated by $\hat{y}_i$ and y_i, respectively. The objective function using L_2 regularization can be written as:

$$\mathcal{L}_{\text{dropout}} = \frac{1}{N} \sum_{i=1}^{N} E(y_i, \hat{y}_i) + \lambda \sum_{l=1}^{L} \left(\|W_l\|_2^2 + \|b_l\|_2^2 \right) \tag{3.1}$$

This formulation is standard in deep learning, where dropout is treated as a regularization technique aimed at reducing overfitting. However, this objective can be reinterpreted in a Bayesian framework. Specifically, training a neural network with dropout is mathematically equivalent to performing approximate variational inference in a deep Gaussian process, where the variational distribution over the weights is implicitly defined by the dropout mechanism.

Under this probabilistic interpretation, the loss function corresponds to the minimization of the variational free energy. This leads to a new training objective, which incorporates both a data-fitting term and a regularization term derived from the Kullback–Leibler divergence between the approximate and true posterior over the model parameters:

$$\mathcal{L}_{\text{GP-MC}} \propto \frac{1}{N} \sum_{n=1}^{N} - \log p(y_n | x_n, \hat{\omega}_n) + \sum_{l=1}^{L} \left(\frac{pl^2}{2\tau N} \|M_l\|_2^2 + \frac{l^2}{2\tau N} \|m_l\|_2^2 \right) \tag{3.2}$$

Here, $\hat{\omega}_n$ denotes a sampled dropout configuration, τ is the model precision (related to the inverse of observation noise), M_l are the variational weight means, and p is the dropout probability. This formulation establishes a direct link between the commonly used dropout regularization in deep learning and a principled Bayesian approximation.

Now the key idea is to perform multiple stochastic forward passes through the model at test time, with dropout active, and to use these samples to approximate the predictive distribution. Given an input x^*, we are interested in the predictive distribution $q(y^*|x^*)$, which incorporates uncertainty over the model parameters. This is defined as:

$$q(y^*|x^*) = \int p(y^*|x^*, \omega)\, q(\omega)\, d\omega \tag{3.3}$$

Here, $q(\omega)$ represents the approximate posterior over the weights induced by the dropout mechanism, and $p(y^*|x^*, \omega)$ is the likelihood of the prediction given a particular weight configuration ω.

Since this integral is intractable, it is approximated using Monte Carlo sampling. By drawing T samples $\{\omega_t\}_{t=1}^T$ from the dropout distribution and computing the corresponding outputs, we obtain a sample-based approximation of the predictive mean:

$$\mathbb{E}_{q(y^*|x^*)}[y^*] \approx \frac{1}{T} \sum_{t=1}^{T} \hat{y}^*(x^*, \omega_t) \tag{3.4}$$

This estimate is referred to as MC Dropout and in practice this is equivalent to performing T stochastic forward passes through the network and averaging the results.

3.3.2 Uncertainty metric extraction method

To estimate uncertainty using the MC Dropout technique, the trained neural network was evaluated multiple times with dropout layers activated during inference. Specifically, 10 stochastic forward passes were performed for each input sample. This approach introduces randomness at each pass, leveraging the dropout-induced variability to approximate a distribution over the network outputs. Than two different pipelines were followed for the segmentation output and the reconstruction output.

Segmentation pipeline

The operations on the various segmentation outputs during MC inference served a dual purpose: it was used both to extract quantitative uncertainty metrics for the

segmentation task, and to construct a conservative mask. This mask was crucial for the subsequent reconstruction pipeline.

- **Mask uncertainty extraction:** To measure uncertainty on the segmentation output, the **normalized entropy measure** was used.

$$\mathbf{H} = -\left[p_r \log p_r + (1 - p_r) \log(1 - p_r)\right] \cdot \frac{1}{\log 2} \in [0, 1] \tag{3.5}$$

  Where p_r is the foreground probability and is obtained by averaging the MC inferences T.

$$p_r = \frac{1}{T} \sum_{t=1}^{T} p_{r,t} \tag{3.6}$$

  Than, a median operation was applied to the resulting matrix to obtain an uncertainty metric on the segmentation. This metric will be referred to as **Mask uncertainty**.

- **Conservative Mask extraction:** This involves creating a final mask that includes all the pixels considered 'important' by the network. Practically, it means that if a pixel was marked as relevant (i.e., set to true) in at least one of the inferences, it will also be set to true in the final conservative mask. This can be translated in a logical OR operation between all of the masks inferenced.

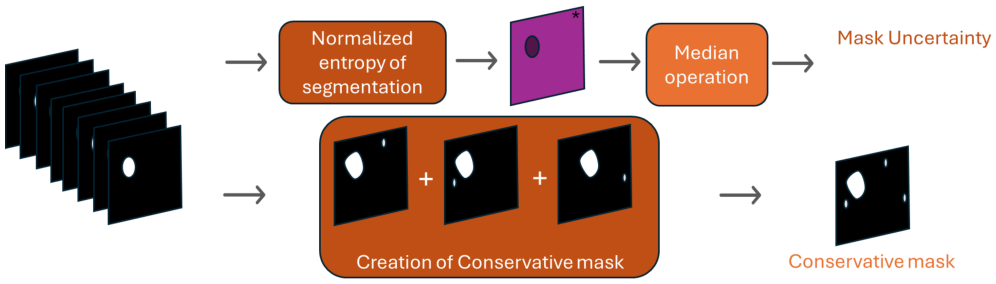


Figure 3.5: Visual scheme of the described segmentation pipeline.
Note: The colors used in this diagram are intended solely for explanatory purposes.

Reconstruction pipeline

To aggregate the results and quantify the predictive uncertainty, either the standard deviation or the variance are computed across the ensemble of predictions for each pixel. This matrix will be referred as the **uncertainty matrix**. Then the relative conservative mask produced in the segmentation pipeline is weighted by some numeric factors. The factor used on the conservative masks were:

- **1.5** for the segmented pixels (the *true* values of the mask);
- **0.5** for the background pixels (the $false$ values).

This operation produces the **weighted mask**, which is then applied to the uncertainty matrix, obtaining the **Weighted Uncertainty on Reconstruction (WUR) matrix**. This is made to weight the uncertainty to make the segmented parts be numerically higher than the background parts, and thus becoming more relevant in the uncertainty metric calculation. After obtaining the WUR matrix, three metrics are calculated:

- **UQmax**: maximum value of the WUR matrix;
- **UQmean**: mean value of the WUR matrix;
- **MWUR**: Median value of the WUR matrix.

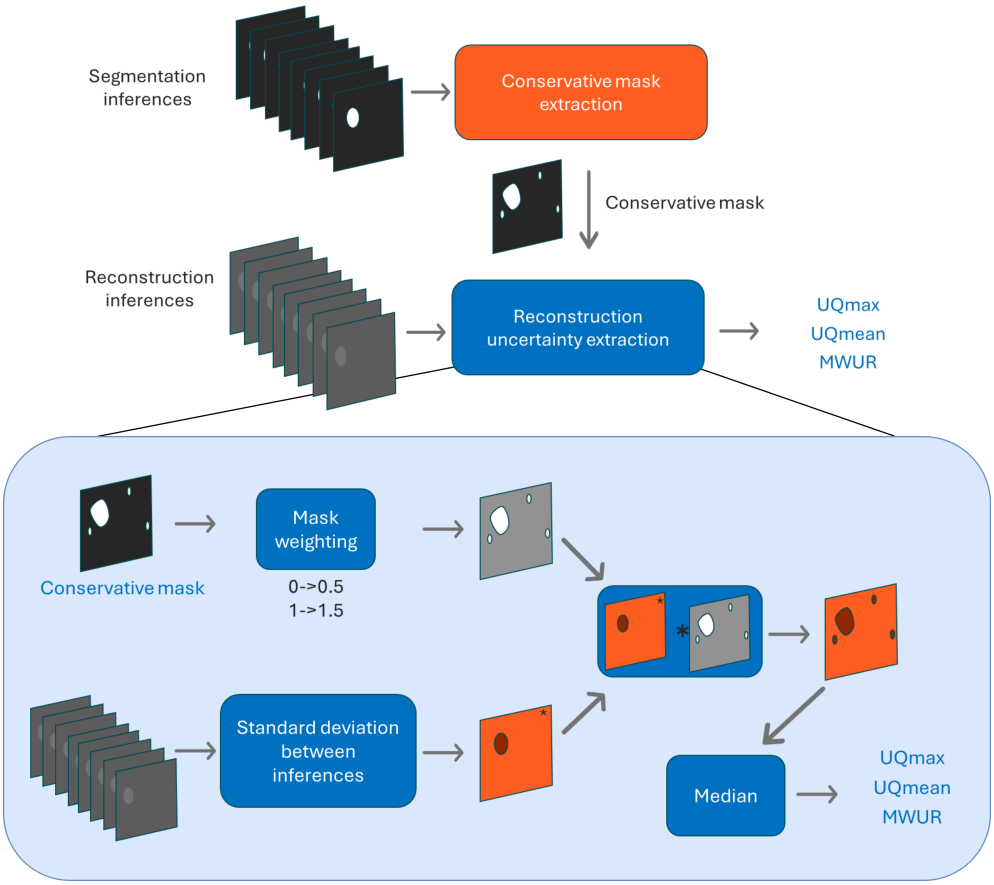


Figure 3.6: Visual scheme of the described Reconstruction pipeline.
Note: The colors used in this diagram are intended solely for explanatory purposes.

3.4 Network training and evaluation

3.4.1 Training parameters

During each of the training the parameters where kept the same to make the results less subject to variability dependent from their variation. The paramteres chosen were the following: The batch size was chosen as the largest value that could fit

Parameter	Value
Learning Rate	0.00005
Batch Size	8
Epochs	50
Segmentation optimizer	Adam
Reconstruction optimizer	Adam
Gamma	0.7

Table 3.1: Training parameters.

within the available GPU memory constraints. Is to be notice that, due to the fact that the network produces both a reconstruction and a segmentation, the parameters of each of the expanging path are optimized separately.

3.4.2 Loss functions

Loss functions play a central role in the training of neural networks by quantifying the discrepancy between the model's predictions and the ground truth. During training, the optimization algorithm adjusts the model's parameters to minimize this loss, thereby improving its performance. In this particular instance 2 different loss function were employed:

- **Dice-Sørensen coefficient (Dice)**[28]: is a statistic used to gauge the similarity of two samples and is commonly employed to evaluate the quality of a segmentation algorithm.

$$\text{Dice}(A, B) = \frac{2|A \cap B|}{|A| + |B|} \qquad \text{Dice} \in [0,1] \tag{3.7}$$

  Where A is the set of predicted positive elements and B is the set of ground truth positive elements. The Dice coefficient measures the similarity between two sets by comparing the size of their intersection to the average size of the sets. A value of 1 indicates perfect overlap between the predicted and actual positive regions, while a value of 0 means there is no overlap at all. This

metric was used as a loss function to guide the training of the segmentation branch.

- **L1 loss**[29]: The L1 loss, also known as Mean Absolute Error (MAE), minimizes the absolute differences between the predicted values and the ground truth. The closer to zero is the L1, the closer the produced reconstruction is similar to the ground truth.

$$\mathcal{L}_{\text{L1}} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{3.8}$$

  where:

  - $\mathcal{L}_{\text{L1}}$ is the L1 loss (MAE),
  - N is the total number of samples,
  - y_i is the true value (ground truth) for the i-th sample,
  - $\hat{y}_i$ is the predicted value for the i-th sample.

  This metric was used as a loss function to guide the training of the reconstruction branch.

3.4.3 Network evaluation metrics

To evaluate the performance of the network, the same metrics used during training were employed, namely the MAE and the Dice coefficient.

3.5 Uncertainty metric evaluation criterion

To evaluate the performances on estimating the uncertainty on the output coming from the model, the uncertainty metrics for the reconstruction were assessed in terms of:

- Correlation with the MAE: Higher uncertainty should reflect into a higher imprecision in the reconstruction, thus the two should show some degree of correlation between each others.

- Ability to differentiate between simulated images and real acquisitions: Real images should exhibit higher intrinsic uncertainty compared to simulated ones, and this difference should be reflected in the uncertainty metric values.

3.6 Empirical Analysis for Metric Selection and Model Tuning

In this section will be illustrated the methods employed to guide the decisions made to construct the best possible "uncertainty evaluation system".

3.6.1 Best Drop out values for Monte Carlo dropout and best uncertainty metric selection

To select the optimal dropout values for use during Monte Carlo inference, several different dropout probabilities were considered. The dropout layers in the network were first divided into three distinct subcategories:

- **Contracting Dropout (c)**: This set includes only the single dropout layer located after the contracting path. This dropout aims to capture the variability from the contracting path.
- **Segmentation Dropout (s)**: This set includes the dropout layers that belong to the segmentation path.
- **Reconstruction Dropout (r)**: This set includes the dropout layers that belong to the reconstruction path.

To evaluate the best combination of dropout values, the resulting uncertainty metrics were assessed with the methods described in the Section 3.5

The dropout probability values for all three categories were initially set to the same value, and seven different values were evaluated: **0.001; 0.01; 0.1; 0.2; 0.3; 0.4; 0.5**. Then, to evaluate the individual impact of dropout variability on the two expanding paths, the following two combinations were tested:

- $[\mathbf{c} = 0.5; \mathbf{s} = 0.1; \mathbf{r} = 0.5]$
- $[\mathbf{c} = 0.5; \mathbf{s} = 0.5; \mathbf{r} = 0.1]$

The evaluation of the best dropout values was also indirectly useful for determining which reconstruction uncertainty metric would be preferred in the following steps.

3.6.2 Uncertainty metric normalization attempts

The uncertainty metrics described in the previous section all suffer from the drawback of not having a defined upper bound. This lack of normalization poses several issues. First, it hinders the interpretability of the uncertainty values, as there

is no clear reference for what constitutes a "high" or "low" uncertainty. Second, it complicates the comparison across different samples, models, or datasets, since the range of possible values may vary significantly. Lastly, it may reduce the robustness of threshold-based decision processes, as fixed thresholds become arbitrary and potentially non-transferable between experiments. To try solve this issue, two different normalization attempts where tested:

- **First normalization attempt**: After obtaining the WUR matrix, the pixels values are normalized by the maximum value of the current matrix.
- **Second normalization attempt**: The maximum pixel value of the WUR matrices obtained inside the construction set (training set + validation set) is kept and than used to normalize the WUR matrices in the whole dataset.

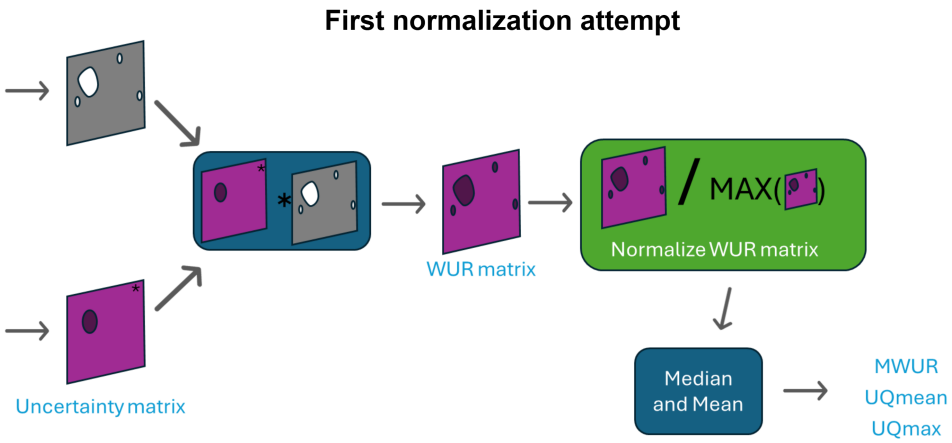


Figure 3.7: Visual scheme of the first normalization attempt.

3.6.3 Uncertainty metric as loss function during training

Once the best uncertainty metric was selected and evaluated as satisfactory to evaluate the reconstruction uncertainty, it was decided to test its potential as a loss function to be used during the training phase for the reconstructions. Two different approaches to insert this new loss where put on trial:

- **Only uncertainty metric**: Only the new loss is calculated and the net adjust itself on the basis of this loss.
- **Weighted loss**: the MAE loss used in the basic training format and the new uncertainty loss are weighted and than

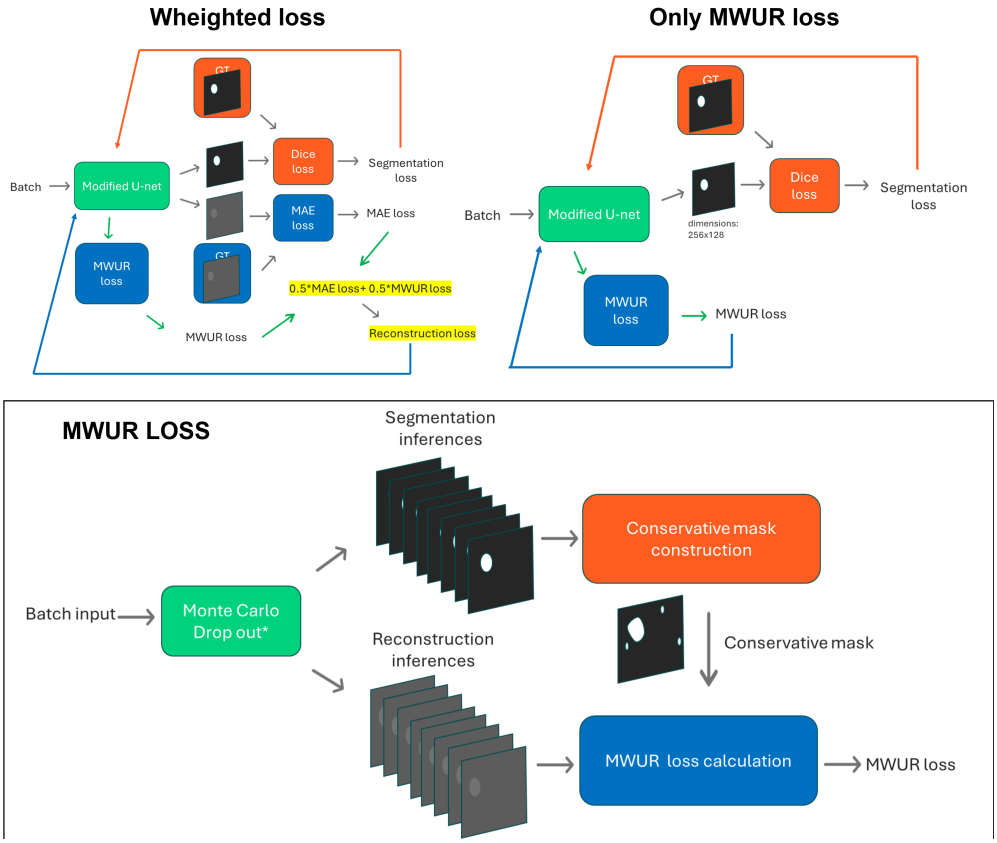


Figure 3.8: Schemes of the losses combination with MWUR metric

3.6.4 Training and evaluation on reduced dataset

To test the consistency of the obtained uncertainty metric, multiple ulterior trainings were performed with reduced dataset. The purpose of reducing the number of data on which the training operation is performed is to obtain a theoretical worsening of the performances, this way is possible to verify that the network result do not influence the ability of the uncertainty metric of differentiating the simulated data from the real data.

Reduced datsets definition

The reduced dataset were obtained by selecting a sub set of the total data; three dataset were obtained:

- **Reduced Dataset** 4_{th}: Dataset is reduced to a fourth of the original number of entries;
- **Reduced Dataset** 8_{th}: Reduced to a eighth of the original number of entries;
- **Reduced Dataset** 16_{th}: Reduced to a sixteenth of the original number of entries.

In the next table are reported the number of the entries present in the training set, validation set and test set fore each sub dataset. The criterion on which the

Set name	**Training set**	**Validation set**	**Test set**
RD4_{th}	1600	200	200
RD8_{th}	800	100	200
RD16_{th}	400	50	50

Table 3.2: Training, validation and test set composition of each subset

original dataset entries were distribuited in each of the reduced dataset's sets to create their own reduced training, reduced validation and reduced test set was to take a sub sets of the entries in their respective original set. In particular the first k elements from each set were taken, with k obtained with the formula 3.9.

$$k = \frac{total\ length\ of\ the\ set}{reduction\ factor} \tag{3.9}$$

This excludes the possibility of presenting different input data between the trainings and thus the possibility of introducing biases on the subsequent evaluations. To evaluate performances on real data, the entirety of the available real data was used for each of the obtained networks.

Training on the reduced training sets

To assure that each network result was comparable with the other, each training was performed using the same training parameters and random seed of the original network employed. Two types of training were conducted on the basis of the loss function applied to the reconstruction branch: one using the standard mae loss, while the other using the previously described weighted loss. One training was performed for each combiantion of loss and reduced dataset for a total of six additional training.

3.7 Additional test of uncertainty metric on photoacustic images

Lastly the metric has been tested on a completly different dataset composed of photoacustic acquisitions.

Photoacustics

Photoacoustic imaging (PAI) is recent biomedical imaging technique that enables detailed visualization of biological structures by combining the optical absorption contrast of light with the deep tissue penetration and high spatial resolution of US. This technique relies on the photoacoustic effect in which pulsed laser light is absorbed by tissue chromophores, causing a rapid and localized temperature rise leading to thermoelastic expansion and the emission of ultrasonic waves. These are then captured and reconstructed into meaningful images.

A typical PAI system includes a laser source, an US transducer, and data acquisition equipment. Together, these components make it possible to perform non-invasive and non-ionizing imaging with excellent spatial resolution and reasonable imaging depth. PAI is especially valuable for visualizing features such as blood vessels, oxygen saturation, and tissue structures, thanks to its sensitivity to endogenous absorbers like hemoglobin or melanin, as well as its compatibility with targeted contrast agents. As a result, photoacoustic imaging shows great promise in a range of biomedical applications, including cancer diagnosis, dermatology, cardiovascular imaging, and neuroscience.[30]

Photoacoustic dataset

The dataset used to tryout the described uncertainty metrics on the task of image reconstruction applied to the photoacoustic data was composed as follows:

- Raw data of size 128x790 px;

- Concentration or Oxygenation maps of size 256x256, which can be compared to the previously described segmentation maps used on the ultrasound image reconstruction;
- Reconstruction model based, which are reconstruction image obtained by commonly used means.

All data of the photoacoustic dataset was obtained from simulated acquisitions.

Training with photoacoustic dataset

A network with the same characteristics of the one employed for the US dataset was used.

Uncertainty metric used on the reconstructions

Only the best performing metric on the US dataset will be used to evaluate the uncertainty of the obtained photoacoustic reconstructions. The metric will be obtained in the same means described in the other sections, using the resulting Concentration maps outputs instead of the segmentation.

Evaluation of the uncertainty on the photoacoustic dataset

To evaluate the performances in terms of uncertainty, the box-plots will not be used due to the fact that only simulated data was available, making the use of box-plot superfluous. The metric will be evaluated strictly in terms of correlation with MAE.

Chapter 4

Results

This section is dedicated to presenting the results obtained following the methods described in in the Chapter 3

4.1 Network Performances

The results obtained after training the network with the basic configuration are presented in Table 4.1.

	MAE	**Dice Coefficient**
Training set	0.692 ± 0.0132	0.9623 ± 0.0599
Validation set	0.696 ± 0.0132	0.9526 ± 0.0785
Test set	0.691 ± 0.0128	0.9509 ± 0.0879
Real set	0.1189 ± 0.0289	Not applicable

Table 4.1: Performance of the network on each dataset.

Since no ground truth was available for the Real set, the Dice coefficient result was omitted. Additionally, attempting to manually segment the images to generate a ground truth would have led to a biased evaluation of the network compared to the other datasets. This is because it would not have been possible to objectively assess what was correctly segmented by the network, as the operator might consider different regions of the image important compared to those emphasized by the network. This discrepancy arises from the fact that the network was trained on a different dataset, potentially leading to divergent interpretations of the relevant anatomical or structural features.

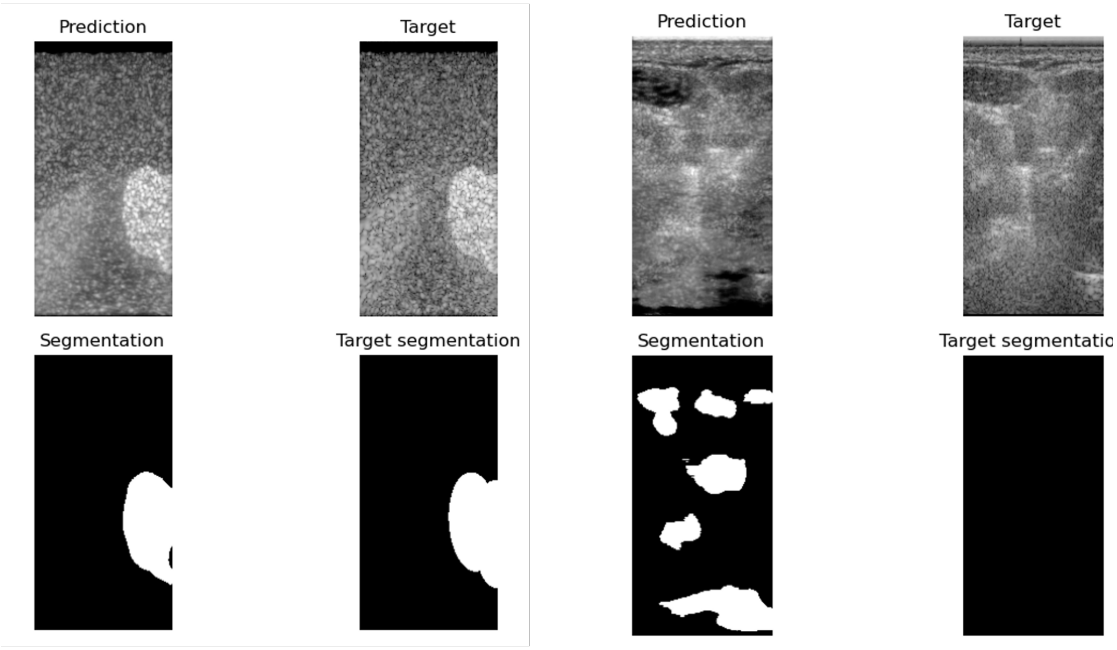


Figure 4.1: On the **left** an example of ground truths and outputs obtained from a simulated acquisition. On the **right** from a real acquisition.

4.2 Results of uncertainty metrics for reconstruction

Each of the previously described uncertainty metrics for the reconstruction output of the network have been tested in terms of correlation with the MAE and on their capability to clearly differentiate between real and simulated images with a box plot. The results are shown in Figure 4.2. The results in terms of the Pearson correlation coefficient, are shown in the Table 4.2

Metric	**Validation set**	**Test set**	**Real data**
MWUR	**corr:** 0.665 **p-value:** 0.0	**corr:** 0.654 **p-value:** 0.0	**corr:** 0.892 **p-value:** 0.001
UQmax	**corr:** -0.033 **p-value:** 0.352	**corr:** -0.049 **p-value:** 0.164	**corr:** 0.129 **p-value:** 0.74
UQmean	**corr:** 0.378 **p-value:** 0.0	**corr:** 0.354 **p-value:** 0.0	**corr:** 0.717 **p-value:** 0.03

Table 4.2: Pearson correlation and p-values of each metric in Validation, Test and Real sets.

In the boxplots, all the metric are capable of visually distinguish between the simulated and real data, with the most apparent difference being visible with the UQmax. Meanwhile UQmax has not shown any correlation with the MAE, with the best Pearson Correlation coefficient being obtained by the MWUR metric.

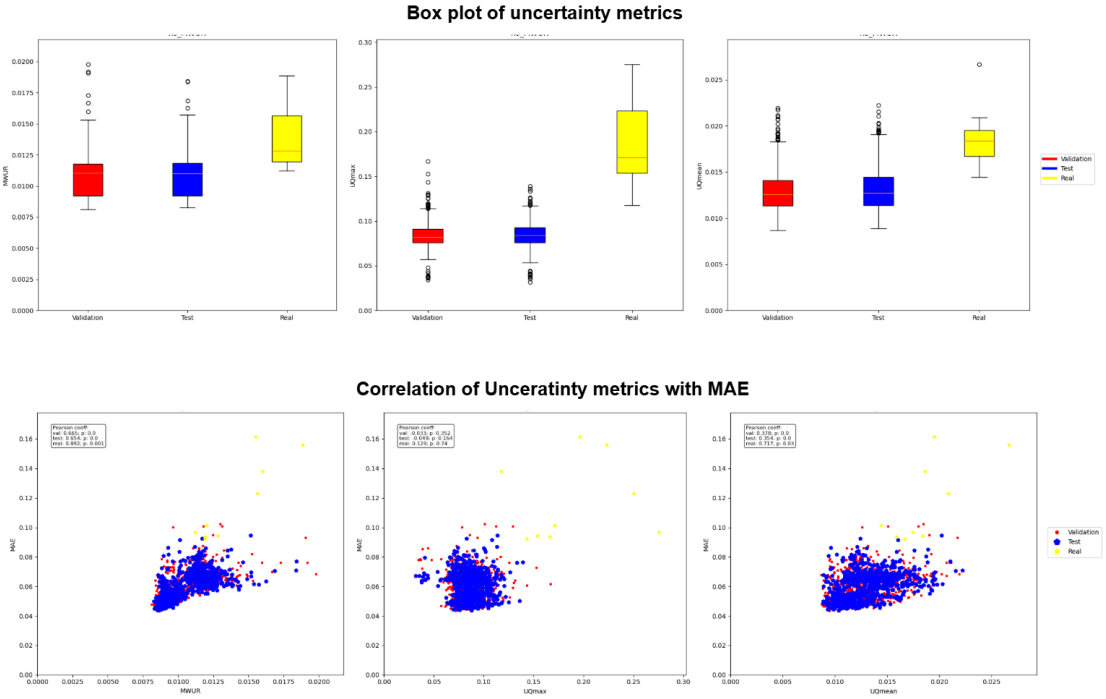


Figure 4.2: In this image, the results of the uncertainty metrics on the terms described in Section 3.5 are shown. In the top part the box plots are shown. In the bottom the correlation graphs between the metric and the MAE.

4.3 Selection of drop out values and selection of best uncertainty metric

Each of the uncertainty metric result on the Validation, Test and Real set was tested at different values of drop out probabilities. From the resulting box plots in Fig 4.3, is apparent that each of the metric grows of values when the drop out probability becomes bigger. Moreover, an increase of dropout probability is translated in more distinguishable boxes between the real data and the simulated data. Metrics calculated using reconstruction branch drop out probability equal to 0.1 and segmentation branch drop out probability equal to 0.5, obtained lower results than the mirrored configuration. In every metric calculated with dropout probabilities higher than 0.2 is possible to distinguish the simulated data from the real data. The resulting plots of the different combinations of metric and drop out values are shown in Fig 4.4. The UQmax (Fig 4.4 A) has not shown any kind of correlation in any of the drop out probabilities combinations. The UQmean (Fig 4.4 B) obtained Pearson correlation values between the values of 0.21 and 0.39 on the simulated data with p-values of 0.0. On the real data the correlation is

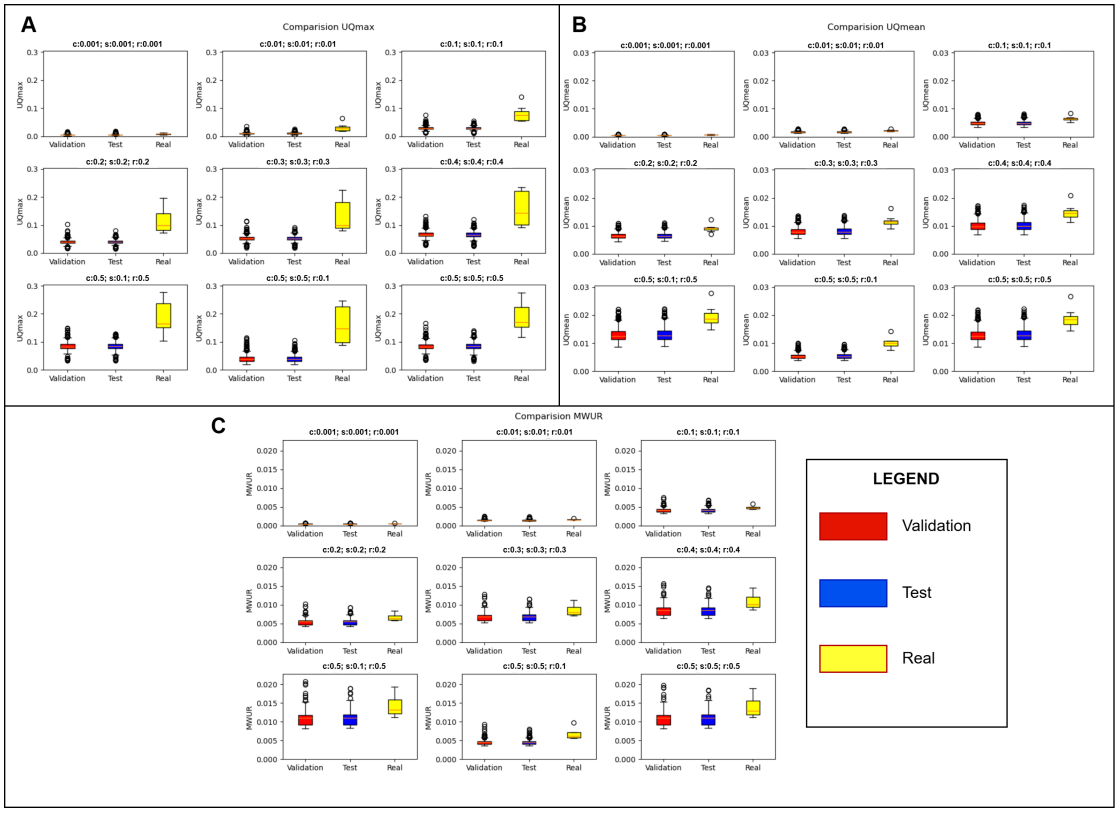


Figure 4.3: Boxplots of each metric at different values of dropout probability.

becomes higher as the drop out probability grows, reaching a value of 0.71 with a p-value of 0.0 in the drop out combination of $[c : 0.5; s : 0.5; r : 0.5]$. Similarly to the UQmean The Pearson correlation coefficient obtained using the MWUR metric grows of magnituide with the increase of dropout probability. The metric reached a correlation of 0.66 and p-value of 0.0 on the combination of $[c : 0.5; s : 0.5; r : 0.5]$. All metrics were able to distinguish between the Real and Simulated data, while the MWUR obtained the best results in terms of correlation. Due to this result, only the MWUR metric was taken into attention in the next experiments. In regards to the dropout probabilities, the best results with each metric where obtained while setting the dropout values of the construction, reconstruction and segmentation path at the value of 0.5. Moving forward this combination of drop out probability values was employed.

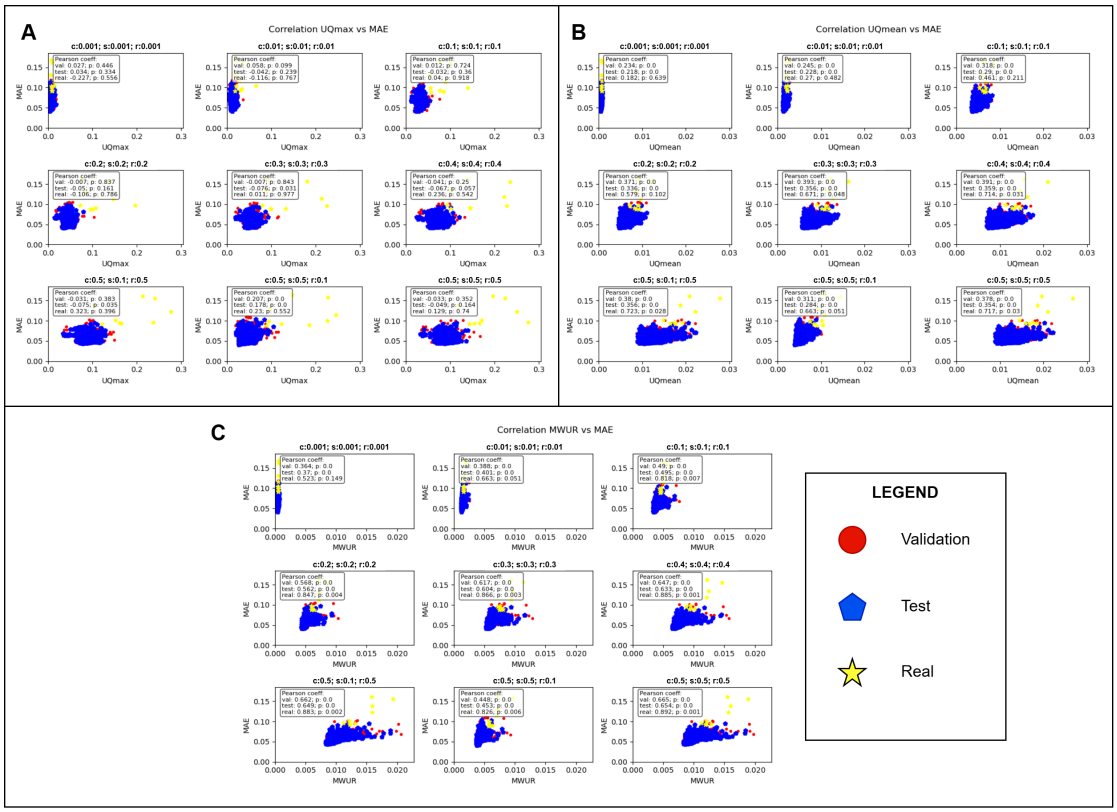


Figure 4.4: Pearson correlation of each metric at different values of dropout probability.

4.4 Results of the normalization attempts

In the Fig 4.5 are shown the results obtained by normalizing the MWUR metric with the first method put on trial. The MWUR metrics obtained during inference were normalized by the maximum value obtained on the WUR matrix. The Pearson correlation coefficient got significantly worse in each of the sets than the not normalized metric. In the box plot, output inferenced became more difficult to neatly separate as the two are almost overlapped. In the Fig 4.6 are shown the

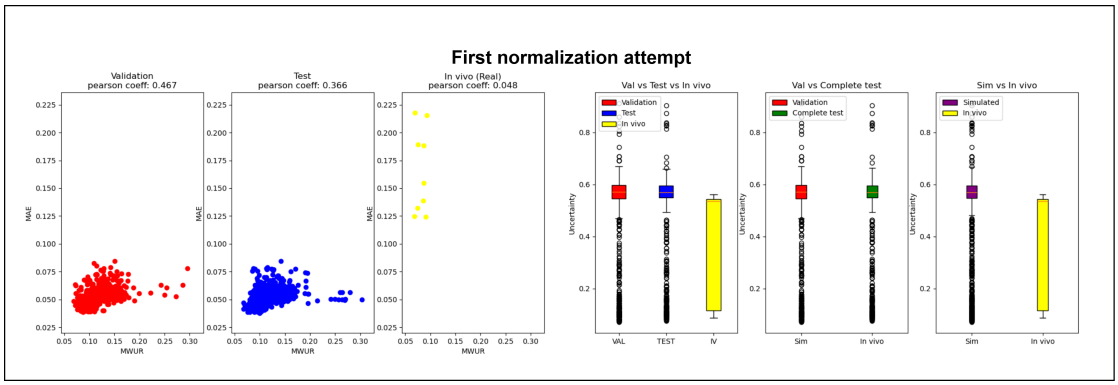


Figure 4.5: MWUR metric normalization by the maximum of WUR matrix

results obtained by normalizing the MWUR metric with the second method. The MWUR metrics obtained during inference were normalized by the maximum value of MWUR obtained on the construction set inferences. The Pearson correlation coefficient did not diverged much from the not normalized metric, even if the results are worse than before. In the box plot, output inferenced stayed distinguishable.

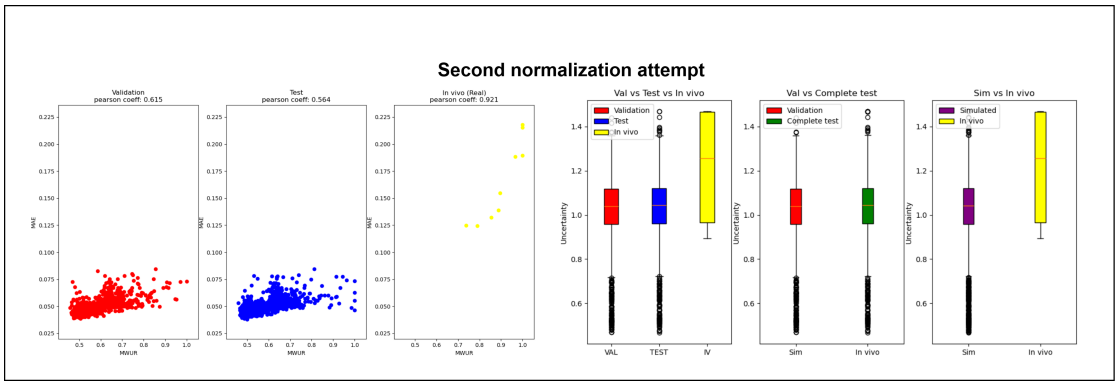


Figure 4.6: MWUR metric normalization by the maximum of MWUR obtained on the construction set

4.5 Performances of the network trained using MWUR as a loss

To gauge the potential of the MWUR metric as a loss function to obtain better network's reconstruction outputs, two additional trainings. The two losses are described in the Section 3.6.3.

Comparison between Dice coefficient results

The Results of the segmentation produced by the networks are in the Table 4.3. The best results were obtained by training the network without using MWUR in any way, while the worst ones were obtained using only the MWUR as a loss. There is not much appreciable difference on the results of the segmentation changing the loss of the reconstruction branch.

Loss	**Training set**	**Validation set**	**Test set**
no MWUR	0.9623 ± 0.0599	0.9526 ± 0.0785	0.9509 ± 0.0879
Both	0.9586 ± 0.0860	0.9516 ± 0.0867	0.9506 ± 0.0976
Only MWUR	0.9502 ± 0.1128	0.9443 ± 0.11499	0.9413 ± 0.1295

Table 4.3: Dice results comparison between the network trained with different losses

Weighting MWUR and MAE loss

The Results of the reconstructions produced by the networks are shown in the Table 4.4 in terms of MAE. The best results on the simulated data were obtained by training the network using the combination of the MWUR loss with the MAE loss even if the ones trained on only the MAE loss are only slightly worse. On the real data the best results were obtained using only the standard loss. The worst results were obtained by using only the MWUR as a loss.

Loss	**Training set**	**Validation set**	**Test set**	**Real data**
no MWUR	0.0692 ± 0.0132	0.0696 ± 0.0132	0.0691 ± 0.0128	0.1189 ± 0.0289
Both	0.0683 ± 0.0124	0.0688 ± 0.0124	0.0681 ± 0.0122	0.1305 ± 0.0334
Only MWUR	0.1915 ± 0.0264	0.1917 ± 0.0259	0.1909 ± 0.0257	0.1780 ± 0.0292

Table 4.4: MAE results comparison between the network trained with different losses

4.6 Performances with reduced datasets

Additional trainings were performed on three reduced dataset, containing respectively a 4_{th}, 8_{th} and a 16_{th} of the full dataset entries. Two trainings were performed on each dataset, one using the MAE loss and the other using the weighted loss. All the trainings were performed keeping the same seed and parameters.

4.6.1 Network performances

The results of the inferences of the network trained are shown in the Table 4.5 in terms of MAE and in Table 4.6 in terms of Dice Coefficient.

Dataset	**Training set**	**Validation set**	**Test set**	**Real data**
Full	0.0692 ± 0.0132	0.0696 ± 0.0132	0.0691 ± 0.0128	0.1189 ± 0.0289
Quarter	0.0832 ± 0.0156	0.0834 ± 0.0146	0.0844 ± 0.0156	0.1348 ± 0.0257
Eight	0.0865 ± 0.0128	0.0884 ± 0.0142	0.0881 ± 0.0136	0.1298 ± 0.0257
Sixteenth	0.0947 ± 0.152	0.0968 ± 0.0139	0.0963 ± 0.0129	0.1248 ± 0.0271

(a) Training performed with MAE loss only

Dataset	**Training set**	**Validation set**	**Test set**	**Real data**
Full	0.0683 ± 0.0124	0.0688 ± 0.0124	0.0681 ± 0.0122	0.1305 ± 0.0334
Quarter	0.0863 ± 0.0161	0.0862 ± 0.0156	0.0875 ± 0.0161	0.1309 ± 0.0277
Eight	0.0864 ± 0.0136	0.0877 ± 0.0143	0.0874 ± 0.0141	0.1237 ± 0.0281
Sixteenth	0.0929 ± 0.0139	0.0955 ± 0.0138	0.0943 ± 0.0124	0.1306 ± 0.0259

(b) Training performed with Weighted loss

Table 4.5: Comparison of MAE performances on the reconstruction inferences between network MAE loss trained (**a**) and Weighted loss trained (**b**) with reduced datasets.

The MAE obtained for the same set between the networks trained with different losses are comparable. The MAE steadily worsen on the simulated datasets as the training dataset is reduced. On Real data using results are comparable even when the training dataset is reduced.
The Dice coefficients obtained for the same set between the networks trained with different losses are comparable. The Dice worsen as the training dataset is reduced. From Table 4.6a it can be seen that when the training is performed with the smallest dataset there is a huge drop in performances, while the same thing does not happen when using weighted loss (Table 4.6b). To make sure that the performances obtained where not the result of using a smaller data to evaluate

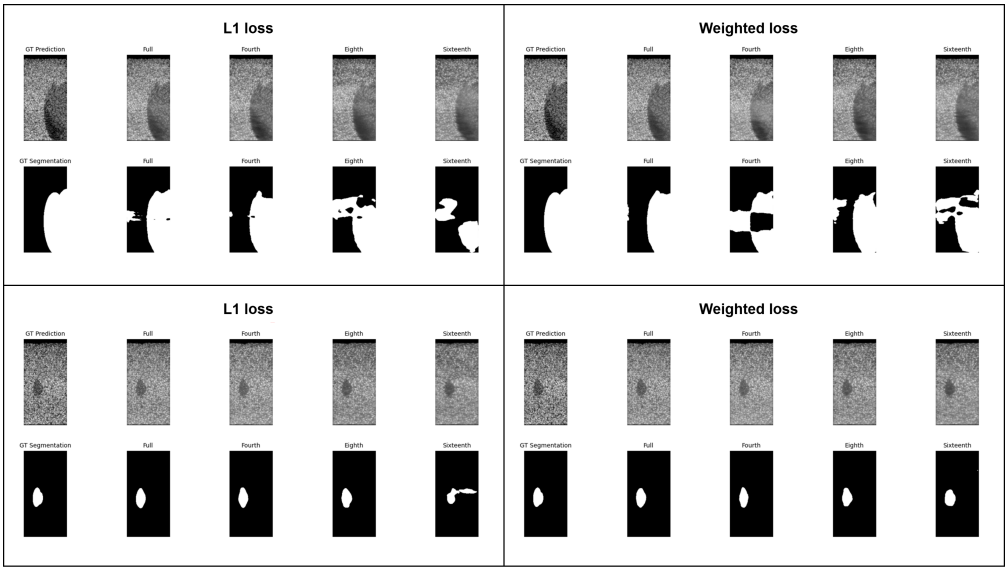


Figure 4.7: Some outputs obtained on the reduced dataset using different losses during training.

Dataset	**Training set**	**Validation set**	**Test set**
Full	0.9623 ± 0.0599	0.9526 ± 0.0785	0.9509 ± 0.0879
Quarter	0.9660 ± 0.0313	0.9378 ± 0.0815	0.9434 ± 0.0994
Eight	0.9221 ± 0.0948	0.8625 ± 0.1868	0.8914 ± 0.1719
Sixteenth	0.6796 ± 0.2985	0.5365 ± 0.2884	0.6414 ± 0.3139

(a) Without Weighted loss

Dataset	**Training set**	**Validation set**	**Test set**
Full	0.9586 ± 0.0860	0.9516 ± 0.0867	0.9506 ± 0.0976
Quarter	0.9536 ± 0.0670	0.9263 ± 0.1071	0.9327 ± 0.1142
Eight	0.9288 ± 0.0869	0.8815 ± 0.1674	0.8891 ± 0.1794
Sixteenth	0.8981 ± 0.1062	0.8322 ± 0.1392	0.8435 ± 0.1901

(b) With Weighted loss

Table 4.6: Comparison of Dice performances on the segmentation inferences between network MAE loss trained (**a**) and Weighted loss trained (**b**) with reduced datasets.

the mertics, was performed a comparison of the performance on the inferences of reduced test set and full test set obtained with the same network; only the two smaller datasets were put to test. From the results shown in Table and Table it is possible to see that performaces remain similare between the full and reduced test set.

Dataset	**Test**	**Full Test**
Eight	0.0881 ± 0.0136	0.0871 ± 0.0131
Sixteenth	0.0963 ± 0.0129	0.0955 ± 0.0146

(a) Training with MAE loss only

Dataset	**Test**	**Full Test**
Eight	0.0874 ± 0.0141	0.0867 ± 0.0136
Sixteenth	0.0943 ± 0.0124	0.0935 ± 0.0135

(b) Training with Weighted loss

Table 4.7: Comparison of MAE performances on reconstruction inferences between networks trained with MAE loss (**a**) and Weighted loss (**b**) on reduced datasets.

Dataset	**Test**	**Full Test**
Eight	0.8914 ± 0.1719	0.8915 ± 0.1399
Sixteenth	0.6414 ± 0.3139	0.6502 ± 0.2968

(a) Training with MAE loss only

Dataset	**Test**	**Full Test**
Eight	0.8891 ± 0.1794	0.8982 ± 0.1339
Sixteenth	0.8435 ± 0.1901	0.8537 ± 0.1533

(b) Training with Weighted loss

Table 4.8: Comparison of Dice coefficient performances on reconstruction inferences between networks trained with MAE loss (**a**) and Weighted loss (**b**) on reduced datasets.

4.6.2 MWUR performances on reduced datasets

The performances of the MWUR metric were evaluated in terms of how distinguishable are the box plot of the simulated dataset from the real data and how correlated is the MWUR obtained in respect to the MAE. The box plot in Fig 4.8 shows that is possible to distinguish between the simulated and real data with the

output of the 2 differently trained network between the simulated set and Real data even when datasets are reduced.
The Pearson correlation plot in Fig 4.9 shows that the correlation degrades as the dataset is reduced for both losses, but that is still possible to appreciate a correlation between the MWUR metric result and the MAE.

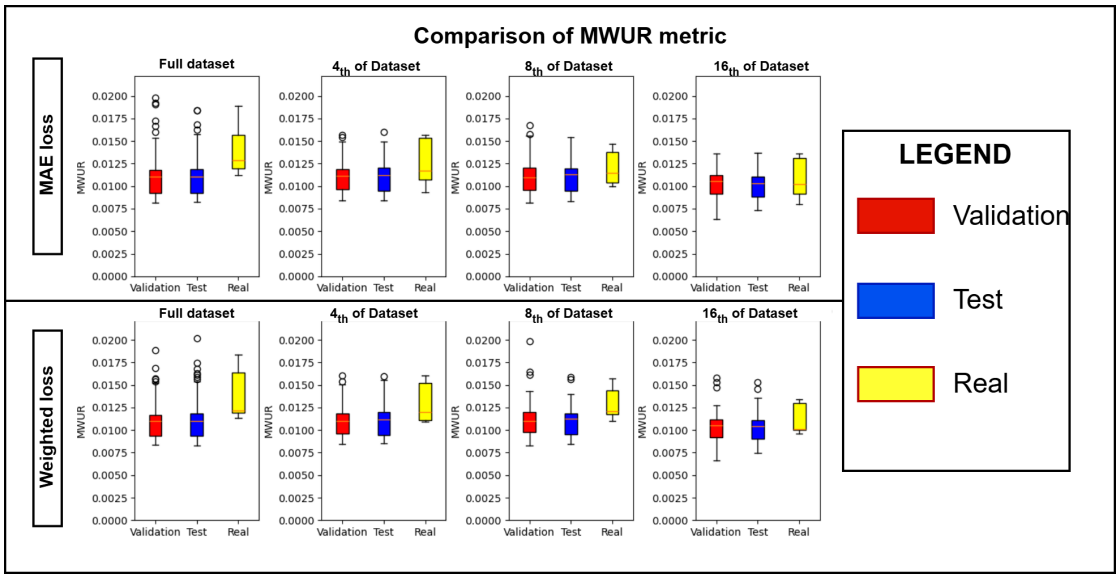


Figure 4.8: Box plots of MWUR metric obtained using reduced datasets. At the Top training with MAE loss, at the Bottom with the Weighted loss.

To verify that results are not due to a smaller pool of the data taken into consideration, further analysis was made on the inferences of the network on the full test for the the 2 smaller reduced datasets. From the Fig. 4.10 and Fig.4.11 it can be appreciated that results are coherent with the ones on the reduced test.

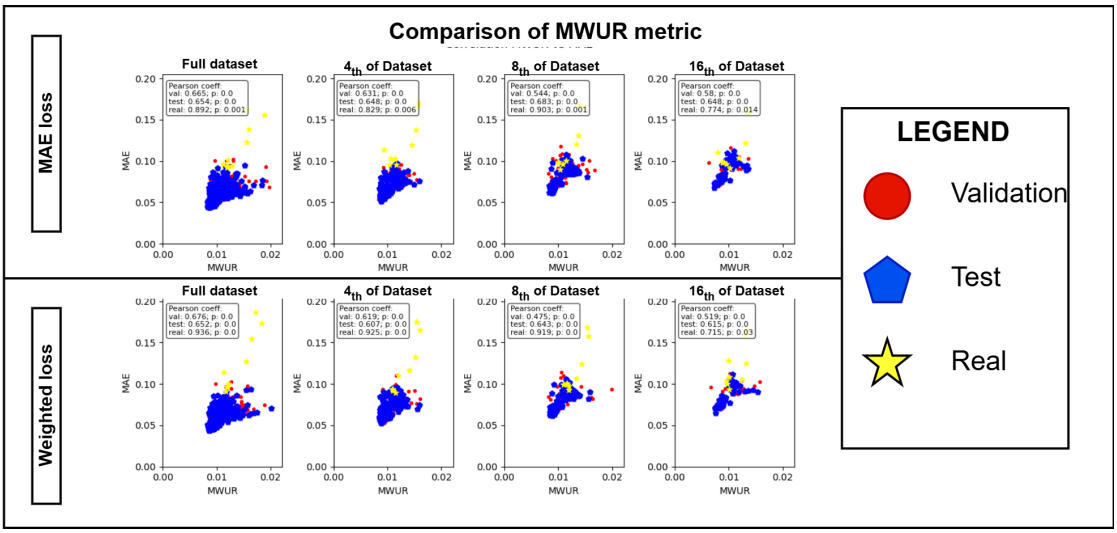


Figure 4.9: Pearson correlation plots of MWUR metric obtained using reduced datasets. At the Top training with MAE loss, at the Bottom with the Weighted loss.

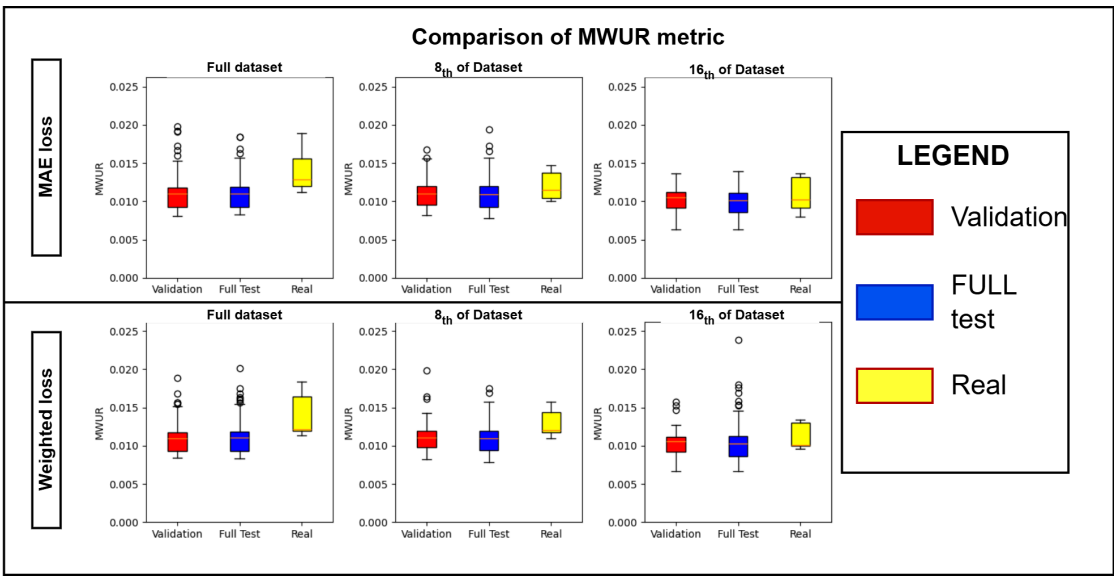


Figure 4.10: Box plots of MWUR metric obtained using reduced datasets with full test. At the Top training with MAE loss, at the Bottom with the Weighted loss.

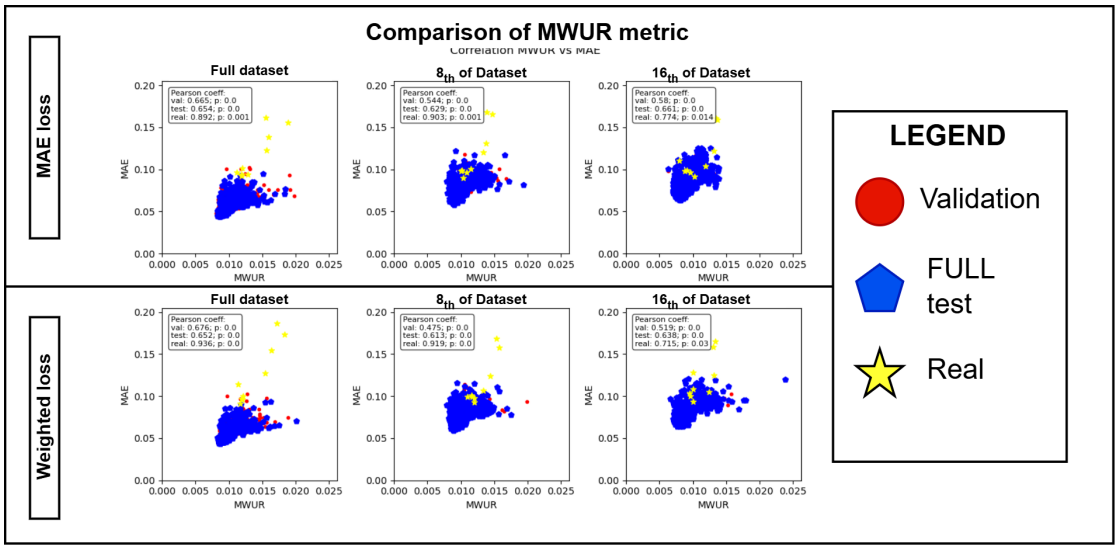


Figure 4.11: Pearson correlation plots of MWUR metric obtained using reduced datasets with full test. At the Top training with MAE loss, at the Bottom with the Weighted loss.

4.7 MWUR performance on the photoacustic acquisitions dataset

The Pearson correlation obtained on the PA dataset was of 0,471 with a p-value of 0.0 on the validation set, while on the test set the results were respectively of 0.142 and 0.03. While the p-value for both set is enough low to consider the existence of a correlation, the coefficient obtained on the test is considerably lower than the one on the validation set. The Figure 4.12 shows the graphical representation of the results.

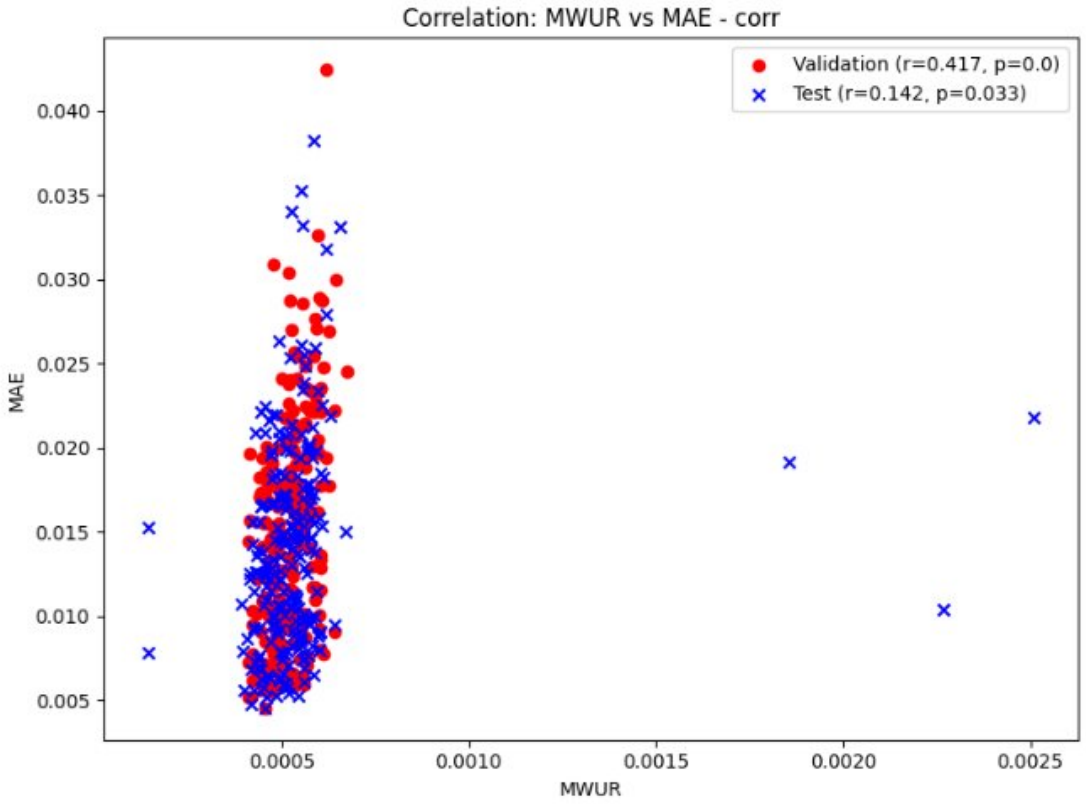


Figure 4.12: Pearson correlation of MWUR against MAE for the PA validation set e test set.

Chapter 5

Conclusions

This thesis aimed to investigate the utility of uncertainty metrics—particularly the MWUR metric—in evaluating the quality of neural network reconstructions and distinguishing real from simulated data. Additionally, the study explored whether MWUR could serve as a loss function to improve the network's training outcomes, particularly in scenarios involving limited data availability.

5.1 Network Performance

The baseline evaluation of the network showed high reconstruction and segmentation performance on synthetic datasets, with low MAE values (0.069) and Dice coefficients above 0.95. However, when applied to real data, a significant drop in performance was observed, with MAE increasing to 0.1189 and no Dice score available due to lack of ground truth. This confirms the network's difficulty in generalizing to real-world data. This was likely due to domain shift between training and real image distributions and also to the low amount of data available coming from real acquisitions.

5.2 Uncertainty Metrics for Reconstruction

Each of the metrics analyzed was able to visually distinguish between the real data acquisition and the simulated ones. This suggests that constructing an uncertainty metric that focuses on the segmented area is a promising direction to estimate the network's uncertainty. Among the uncertainty metrics analyzed, MWUR demonstrated the strongest correlation with MAE (up to 0.892 on real data), confirming its effectiveness in estimating reconstruction error. UQmean showed moderate correlation, while UQmax did not correlate with MAE at all. This difference in correlation may be attributed to the inherent variability of UQmax and

UQmean, as they are computed from the distribution of uncertainty values across the image: UQmean represents the average uncertainty, which can be strongly influenced by small fluctuations across the entire region, while UQmax corresponds to the single highest value in the distribution, making it extremely sensitive to outliers or local spikes. As a result, both metrics tend to be less stable and less representative of the overall reconstruction error compared to MWUR, which is based on the median value , making it more robust and less sensitive to extreme values, which contributes to its higher stability and stronger correlation with the actual reconstruction error.

5.3 Dropout Probability and Best Metric Selection

Increasing dropout probability improved the separability between simulated and real images, especially with higher dropout rates (0.5) in all branches. Moreover, MWUR consistently provided the best correlation with MAE across dropout settings, outperforming both UQmean and UQmax. Based on these findings, MWUR was selected as the preferred metric for subsequent analysis, and the dropout configuration [c: 0.5; s: 0.5; r: 0.5] was adopted as optimal. Particularly, the experiments suggest that a high dropout probability on the reconstruction branch plays a more critical role than on the segmentation branch. This is evidenced by the fact that the configuration [c: 0.5; s: 0.1; r: 0.5] showed significantly better separability and correlation across all metrics compared to [c: 0.5; s: 0.5; r: 0.1]. This result is expected, as all the metrics are derived by weighting the reconstruction uncertainty using the segmentation, but their core is still based on the reconstruction uncertainty itself. Therefore, increasing the dropout in the reconstruction branch leads to greater variability.

5.4 Normalization of MWUR

Two normalization strategies were tested to standardize the MWUR values. The first normalization method was completely ineffective and actually harmed the metric's ability to distinguish between different types of data. This outcome was expected, as normalizing based on the full uncertainty matrix causes the median values to become very similar across samples, thereby reducing the variability that is essential for the metric to differentiate between real and simulated data.
The second approach showed better results, as it was able to preserve both the separability between real and simulated data and the correlation with the MAE. However, its main limitation lies in the fact that the normalization was performed

by scaling the MWUR values based on the maximum obtained on the available construction set. This introduces several potential issues:

- First, it makes the normalization dataset-dependent, meaning that results may not generalize well when the model is applied to different data distributions or in real-world scenarios where the same construction set is not available.

- Second, the maximum value used for normalization could be sensitive to outliers or rare cases within the construction set, introducing instability in the normalization factor.

- Lastly, the need to normalize by the maximum value of the MWUR implies that the value obtained on the construction set is always required, which makes it impractical to employ the metric as a loss function during training. Since the normalization depends on an external reference computed post hoc, it cannot be dynamically integrated into the optimization process, thus preventing its use as a real-time loss term within the learning framework.

The identified issues have led to the conclusion that non-normalized MWUR values provide better interpretability and performance, and therefore should be used as such in practical applications.

5.5 Training with MWUR as a Loss Function

When MWUR was used alone as a loss function, both reconstruction and segmentation performances degraded substantially. In contrast, combining MWUR with MAE yielded slightly improved reconstruction on synthetic data, but worse performance on real data compared to using MAE alone. These results suggest that MWUR cannot be used a standalone loss function as it cannot apparently measure and take into account all the information needed to produce quality images form raw data. Instead its use in combination with the MAE loss to construct a weighted loss could be able to take into consideration factors that are not taken into account with the basic loss, leading to benefits in the final reconstruction. The most significant issue with using the MWUR as a loss function in any form is that its calculation requires performing Monte Carlo inference during the training phase, resulting in a training time increase of more than double compared to training the network without it. So even if the training with the weighted loss obtain better results, the improvement is not of sufficient entity to justify its use instead of the MAE loss.

5.6 Training with Reduced Datasets

Reducing the size of the training dataset led to progressive degradation in MAE and Dice scores, especially when using only 1/16 of the data. Nonetheless, models trained with the weighted loss (MWUR + MAE) performed slightly better in segmentation tasks than those trained with MAE alone, particularly under data scarcity. This indicates that MWUR may act as a regularizer when training with small datasets, helping to preserve generalization ability. The previously noted significant increase in training time was mitigated by using smaller training sets, but remained noticeable even at a reduced scale.

5.7 MWUR on Reduced Datasets

MWUR maintained its ability to differentiate real from simulated data even when the network was trained on reduced datasets. Though the Pearson correlation with MAE diminished with less training data, it remained statistically significant. The consistency of results on full and reduced test sets further confirmed the robustness of MWUR as a reliable uncertainty metric, even under sub optimal training conditions, suggesting that the previously observed ability to differentiate the simulated from the real data was not the result of a substantial difference in performance between the two sets.

5.8 MWUR on the PA dataset

The results on the correlation coefficient suggest that a correlation between MAE and MWUR exist even if the registered coefficients are considerably lower that the ones obtained on the US dataset. Especially on unseen data, the metric seems to be much less correlated, this could be caused from the fact that the oxigenation maps, even if akin to segmentations, are not the exact same. Regardless the MWUR metric showed potential to be possibly used on reconstruction of data that is not strictly derived from US.

5.9 Final Remarks

The aim of this thesis work was to construct a uncertainty metric that was able to effectively measure or estimate the uncertainty on the reconstructions of US produced by a deep learning models. By leveraging on a modified U-net capable of producing from raw US acquisition both the US image reconstruction and its corresponding segmentation mask, and by performing on it a Monte Carlo inference,

a metric that uses the two outputs to weight the uncertainty on the reconstruction with its corresponding conservative mask was defined. This metric was denominated Median of Weighted Uncertainty on the Reconstruction (MWUR). The MWUR was capable of consistently distinguish between outputs derived by real acquisitions from ones obtained by simulated ones. It also has shown a substantial correlation with the Mean Absolute Error on the reconstruction, suggesting that the metric could be used to understand the quality of the output images. The potential of the metric as a loss function was also explored, resulting in a slight increase in the quality if the reconstruction when combined with the commonly used MAE loss. The previously described ability to distinguish real and simulated data was kept even when performances where purposely worsen by showing the network a smaller pool of data during the training phase. Also, when dataset where reduced to a eighth of the original, the network trained with the weighted loss achieved better segmentation results compared that the one trained using only the MAE loss. MWUR showed some capability to be used on PA dataset different from US acquisition, suggesting that, as long that some form of segmentation is produced or given, is possible to estimate the uncertainty of images derived from US acquisitions. While the MWUR showed promise as an uncertainty metric, it is also important to highlight its shortcomings. The MWUR is extremely dependent on the fact that the employed network is capable of producing both a reconstruction and a segmentation from the same input, this makes it not very flexible to its use with other network configurations, especially when used as a loss function during training. The pool of real acquisitions available was extremely lower compared to the simulated ones, so even if the MWUR was consistently capable of distinguish between the two even when the simulated data was purposely made smaller, it cannot be excluded that its abilities where in part caused by the specific pool of real acquisition given. When the network is trained using the MWUR as a loss function in any form, it leads to an enormous increase in training time and this could not be justified by the only slight increase in the performances shown when it was employed.

In conclusion, even with its shortcomings, the MWUR metric has shown potential to be used as a valid uncertainty metric for quantify the uncertainty of US image reconstruction from raw data acquisitions inputs and showed potential even when applied to the task of PA image reconstruction. In the future the proposed metric could be explored more profoundly, by fine tuning some of its shortcomings and could also be put on trial on dataset that are note derived from some form of US acquisition. The results shown in this work can be seen as a starting point to find a better and more flexible metric in the still young and unexplored field of uncertainty quantification on the reconstruction of US images.

Bibliography

[1] Jørgen Arendt Jensen. «Medical ultrasound imaging». In: *Progress in Biophysics and Molecular Biology* 93.1 (2007), pp. 153–165. DOI: 10.1016/j.pbiomolbio.2006.07.025 (cit. on p. 1).

[2] Nimrod M. Tole. *Basic Physics of Ultrasonic Imaging*. Geneva: World Health Organization, 2005 (cit. on p. 2).

[3] Zixia Zhou, Yuanyuan Wang, Jinhua Yu, Yi Guo, Wei Guo, and Yanxing Qi. «High spatial–temporal resolution reconstruction of plane-wave ultrasound images with a multichannel multiscale convolutional neural network». In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 65.11 (2018), pp. 1983–1996. DOI: 10.1109/TUFFC.2018.2865504 (cit. on p. 3).

[4] Ahmet M. Elbir, Kumar Vijay Mishra, Sergiy A. Vorobyov, and Robert W. Heath. «Twenty-Five Years of Advances in Beamforming: From Convex and Nonconvex Optimization to Learning Techniques». In: *IEEE Signal Processing Magazine* 40.4 (2023), pp. 118–131. DOI: 10.1109/MSP.2023.3262366 (cit. on p. 3).

[5] Libertario Demi. «Practical guide to ultrasound beam forming: Beam pattern and image reconstruction analysis». In: *Applied Sciences* 8.9 (2018), p. 1544. DOI: 10.3390/app8091544 (cit. on p. 3).

[6] Vincent Perrot, Maxime Polichetti, François Varray, and Damien Garcia. «So you think you can DAS? A viewpoint on delay-and-sum beamforming». In: *Ultrasonics* 111 (2021), p. 106309. DOI: 10.1016/j.ultras.2020.106309 (cit. on p. 4).

[7] Deepak Jakhar and Ishmeet Kaur. «Artificial intelligence, machine learning and deep learning: definitions and differences». In: *Clinical and Experimental Dermatology* 45.1 (Jan. 2020), pp. 131–132. DOI: 10.1111/ced.14029 (cit. on p. 6).

[8] Keiron O'Shea and Ryan Nash. «An Introduction to Convolutional Neural Networks». In: *arXiv preprint arXiv:1511.08458* (2015), pp. 1–11 (cit. on p. 7).

[9] Miao Chu, Peng Wu, Guanyu Li, Wei Yang, Juan Luis Gutiérrez-Chico, and Shengxian Tu. «Advances in Diagnosis, Therapy, and Prognosis of Coronary Artery Disease Powered by Deep Learning Algorithms». In: *JACC: Asia* 3.1 (2023), pp. 1–14. DOI: 10.1016/j.jacasi.2022.12.005 (cit. on p. 8).

[10] Massimo Salvi, Silvia Seoni, Andrea Campagner, Arkadiusz Gertych, U. Rajendra Acharya, Filippo Molinari, and Federico Cabitza. «Explainability and uncertainty: Two sides of the same coin for enhancing the interpretability of deep learning models in healthcare». In: *International Journal of Medical Informatics* 197 (Feb. 2025), p. 105846. DOI: 10.1016/j.ijmedinf.2025.105846 (cit. on p. 8).

[11] Silvia Seoni, Vicnesh Jahmunah, Massimo Salvi, Prabal Datta Barua, Filippo Molinari, and U. Rajendra Acharya. «Application of uncertainty quantification to artificial intelligence in healthcare: A review of last decade (2013–2023)». In: *Computers in Biology and Medicine* 165 (Nov. 2023). Available online 1 September 2023, p. 107441. DOI: 10.1016/j.compbiomed.2023.107441 (cit. on p. 9).

[12] Martin Magris and Alexandros Iosifidis. «Bayesian Learning for Neural Networks: an algorithmic survey». In: *arXiv preprint arXiv:2211.11865* (Jan. 2023). Preprint under review (cit. on p. 9).

[13] Yasser Fouad et al. «Deep learning for ultrasound image denoising». In: *Ultrasonics* 108 (2020), p. 106127 (cit. on pp. 10, 11).

[14] Adam C. Luchies and Brett C. Byram. «DNN beamforming for high contrast targets in the presence of reverberation clutter». In: *Proceedings of IEEE International Ultrasonics Symposium (IUS)*. 2020 (cit. on pp. 10, 11).

[15] Bram Luijten et al. «Adaptive ultrasound beamforming using deep learning». In: *IEEE Transactions on Medical Imaging* 39.11 (2020), pp. 3339–3349 (cit. on p. 10).

[16] Sobhan Goudarzi and Hassan Rivaz. «Deep reconstruction of high-quality ultrasound images from raw plane-wave data: A simulation and in vivo study». In: *Ultrasonics* 125 (2022), p. 106778 (cit. on pp. 10, 11).

[17] Elay Dahan and Israel Cohen. «Deep-learning-based multitask ultrasound beamforming». In: *Information* 14.10 (2023), p. 582 (cit. on pp. 10, 11).

[18] Ilya Mamistvalov et al. «Deep-learning-based adaptive ultrasound imaging from sub-Nyquist channel data». In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 69.12 (2022), pp. 3576–3587 (cit. on pp. 10, 11).

[19] Hassan Rivaz et al. «Deep learning to obtain simultaneous image and segmentation outputs from a single input of raw ultrasound channel data». In: *Medical Image Analysis* 73 (2022), p. 102157 (cit. on pp. 10, 11).

[20] Archana Nair et al. «A deep learning-based alternative to beamforming ultrasound images». In: *IEEE International Ultrasonics Symposium (IUS)*. IEEE. 2018, pp. 1–4 (cit. on p. 11).

[21] Archana Nair et al. «A generative adversarial neural network for beamforming ultrasound images». In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 67.11 (2020), pp. 2309–2321 (cit. on p. 11).

[22] Bahareh Haji-Saeed et al. «Artifact suppression for passive cavitation imaging using U-Net CNNs with uncertainty quantification». In: *IEEE Transactions on Medical Imaging* 40.6 (2021), pp. 1635–1647. DOI: 10.1109/TMI.2020.3037793 (cit. on p. 12).

[23] Jørgen A. Jensen. *FIELD: A Program for Simulating Ultrasound Systems*. Technical University of Denmark. Lyngby, Denmark, 1996. URL: http://field-ii.dk/ (cit. on p. 14).

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. «U-Net: Convolutional networks for biomedical image segmentation». In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Vol. 9351. 2015, pp. 234–241 (cit. on p. 16).

[25] Archana Nair et al. «Deep Learning to Obtain Simultaneous Image and Segmentation Outputs From a Single Input of Raw Ultrasound Channel Data». In: *Proceedings of IEEE International Ultrasonics Symposium (IUS)*. 2018 (cit. on p. 17).

[26] Yarin Gal and Zoubin Ghahramani. «Dropout as a Bayesian approximation: Representing model uncertainty in deep learning». In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. Vol. 48. 2016, pp. 1050–1059 (cit. on p. 19).

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. «Dropout: A Simple Way to Prevent Neural Networks from Overfitting». In: *Journal of Machine Learning Research* 15 (2014). Submitted November 2013; Published June 2014, pp. 1929–1958 (cit. on p. 19).

[28] Daniel Bell and Candace Moore. «Dice similarity coefficient». In: *Encyclopedia of Medical Devices and Instrumentation* (2020). DOI: 10.53347/rID-75056 (cit. on p. 23).

[29] Scott Pesme and Nicolas Flammarion. «Online robust regression via SGD on the L1 loss». In: *CoRR* abs/2007.00399 (2020) (cit. on p. 24).

[30] Huibin Liu, Xiangyu Teng, Shuxuan Yu, Wenguang Yang, Tiantian Kong, and Tangying Liu. «Recent advances in photoacoustic imaging: Current status and future perspectives». In: *Micromachines* 15.8 (2024), p. 1007. DOI: 10.3390/mi15081007 (cit. on p. 29).