Deep learning algorithms to improve bone scintigraphy

Akos Kovacs

PhD dissertation

supervisors:
Dr. András Horváth,
Dr. Tamás Bükki

Roska Tamas Doctoral School of Sciences and Technology
Pázmány Péter Catholic University

Budapest, 2023

Abstract

Images taken with a gamma camera typically have a low signal-to-noise ratio and are subject to significant Poisson noise. In this thesis a neural network based noise filter is proposed that can be used with planar bone scintigraphy recordings at multiple noise levels, instead of developing a separate network for each noise level. In addition, a new type of loss function is presented, that is able to take topographical aspects into account.

The proposed denoising solution is a convolutional neural network (CNN) inspired by U-NET architecture. A total of 1215 pairs of anterior and posterior patient images were available for training and evaluation during the analysis. The noise-filtering network was trained using bone scintigraphy recordings with real statistics according to the standard protocol, without noise-free recordings. The resulting solution proved to be robust to the noise level of the images within the examined limits.

During the evaluation, the performance of the networks was compared to Gaussian and median filters and to the Block-matching and 3D filtering (BM3D) filter. My presented evaluation method in this thesis does not require noiseless images and I measured the performance and robustness of my solution on specialized validation sets.

It has been shown that particularly high signal-to-noise ratios can be achieved using noise-filtering neural networks (NNs), which are more robust than the traditional methods and can help diagnosis, especially for images with high noise content.

- Two neural network based noise filters have been designed
- They can be used with planar bone scintigraphy recordings at multiple noise levels
- They were trained on acquisitions created by the standard protocol
- The training and the evaluation method presented in this thesis does not require noiseless images

- The performance and robustness was measured on specialized validation datasets

A common indication for bone scintigraphy is to detect and track bone metastases of various tumours, so as a further step, my team started to develop a software to search for pathological enrichment. One method to create a software component for abnormal enrichment detection and prediction is the use of convolutional neural networks. Related to this topic, we have developed a new segmentation metric, wave loss.

We need a well-defined error function for training neural networks that can solve segmentation problems successfully. In the most common approaches, typically only region-based differences are considered, while the topology, meaning the spatial distribution of pixels, is not taken into account. Our brain can compare complex objects with ease and considers both pixel level and topological differences simultaneously. Comparison between objects requires a properly defined metric that determines similarity between them considering changes both in shape and values. In past years, topographic aspects were incorporated in loss functions where either boundary pixels or the ratio of the areas were employed in difference calculation. I have developed these ambitions further, and I have demonstrated that incorporating topological information in the loss function can improve the segmentation accuracy of various architectures.

During my work I showed how the application of this topographic metric, called wave loss, has increased segmentation accuracy by 3% on both the Cityscapes and MS-COCO datasets, using various network architectures.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Overview of the thesis 3

2 Background and theory **5**
2.1 Overview of the imaging hardware for nuclear medicine imaging . 5
2.2 Overview of bone scintigraphy 9
2.2.1 Mathematical formulation of noise filtering 13
2.2.2 Traditional noise filtering solutions for bone scintigraphy . 13
2.3 Deep learning 14
2.3.1 Convolutional neural networks 16
2.3.2 Training neural networks for segmentation tasks 16

3 Proposed deep learning based noise filtering **19**
3.1 Training strategy 20
3.2 Neural network architecture 22
3.3 Materials and methods 23
3.3.1 Characteristics of the clinical data used 23
3.3.2 Detailed description of neural network architectures 24
3.4 Evaluation and results 27
3.4.1 Comparing the performance of different neural networks and conventional noise-filtering solutions 30
3.4.2 Testing the robustness of the best performing neural network on specialized validation sets 32
3.4.3 Preliminary clinical evaluation 39

3.5 Conclusion 43

4 Proposed loss function for neural network based segmentation: wave loss 45

4.1 Comparison of Shapes and the Binary Wave Metric 45
4.2 Wave loss: Extension of the Wave Metric to Three Dimensions . . 50
4.3 Materials and Methods 55
4.3.1 Simple Dataset for Segmentation 55
4.3.2 Competitor loss functions 58
4.4 Results 59
4.4.1 Semantic segmentation on Cityscapes 59
4.4.2 Instance segmentation on MS-COCO 60
4.4.3 Implications of my finding regarding the wave loss function 62
4.5 Conclusions 63

5 Discussion 65

5.1 Utilization as a Fast Localizer 65
5.2 Utilization as a diagnostic noise filter 67
5.3 The development of a physician-assisting diagnostic solution . . . 67

6 Summary 69

6.1 New scientific results 70

References 87

List of Figures

3.1 Possible artifacts when using MAE as loss function 22
3.2 Denoising quality: MSE vs. MAE 23
3.3 U-NET structure 24
3.4 Losses during training 27
3.5 Evaluation pipeline 30
3.6 Example images from the evaluation pipeline 31
3.7 RMSE performance of the filters 36
3.8 Box plots about the performance of the filters on normal statistics 37
3.9 Box plots about the performance of the filters on 1/3 statistics . . 37
3.10 MAE performance of the filters 38
3.11 Sample image from L-NN 40
3.12 Sample images from L-NN on reduced statistics 41

4.1 Problem with the Hamming distance 46
4.2 Problem with the Hausdorff distance 47
4.3 Illustration of the wave propagation 50
4.4 Wave propagation 54
4.5 Example images from the CLEVR dataset 56
4.6 Losses on the CLEVR dataset 57
4.7 Example images at different iterations from the CLEVR dataset . 58
4.8 Example results on the COCO dataset segmented with Mask-RCNN 61

5.1 Fast localizer sample image 66

List of Tables

3.1 Details of the neural network architecture 25
3.2 RMSE performance of the filters 33
3.3 SSIM performance of the filters 34
3.4 MAE performance of the filters 35
3.5 RMSE performance of NN-based filters on different validation sets. 35
3.6 MAE performance of the filters 36

4.1 Performance on the Cityscape dataset 60
4.2 Performance on the COCO dataset 62

Chapter 1

Introduction

1.1 Motivation

Medical imaging devices are extremely important in today's patient-centered care based on scientific results. These appliances have completely changed the way medicine is practised over the last 30 years. They have made it possible to identify diseases at an early stage (especially in screening tests) and, thanks to early treatment, patients' chances of recovery have improved. The use of X-ray, ultrasound, CT, MRI and other imaging technologies has become commonplace in the clinic and thus widely known in society. Medical imaging is particularly beneficial in detecting and identifying cancer because early detection means better chances of cure. One of the tools routinely used for this purpose is the gamma camera, which uses a variety of radioactive isotopes to obtain information about physiological processes in the body.

I have been working on the improvement of one of the most commonly performed examinations, bone scintigraphy, a common, relatively inexpensive and widely available technique, which is invaluable in the diagnostic evaluation of many pathological conditions due to its sensitivity. In Hungary, an average of 5-600MBq of MDP isotope activity is administered to the patient and a 15-20 minute scan with the device is required. Gamma camera images typically have a low signal-to-noise ratio and are subject to significant Poisson noise. These circumstances motivated me to develop an image enhancement device during my doctoral studies that would allow reducing the injected activity (radioactive dose)

and shortening the imaging time.

Exceptionally high signal-to-noise ratios can be achieved by using noise filtering neural networks (NNs), which can be configured and trained to act as specialised noise filters. An important characteristic of NN-based noise filters is that they learn on noisy input data and also on reference or ground-truth image pairs with significant noise. It has been shown that, nevertheless, a properly constructed NN filter trained in this way can synthesize filtered images with a better signal-to-noise ratio than that of reference images, and outperforms conventional noise filters, e.g. BM3D method, for either Gaussian or Poisson noise [1]. Considering that the image database used for filter training usually consists of a highly limited number of images due to the difficult availability of real patient data, we consider it particularly important to investigate the robustness of the trained NN-based image processing algorithm, i.e., its sensitivity to the noise content of the images and their distribution according to various aspects, including patient age, gender, body mass index value, and the nature and distribution of characteristic pathological structures in the image. This analysis can reveal the robustness of such an image processing algorithm, either on its own or as part of a larger CADx (computer aided diagnosis) system deployed in clinics around the world.

A common indication for bone scintigraphy is to detect and track bone metastases of various tumours, so as a further step, we started to develop a software to search for pathological enrichment. This tool will also allow quantification of the impact of the image enhancement tool on diagnostics, so that an application-specific lesion-based evaluation can be performed in addition to image-based metrics. One method to create a software component for abnormal enrichment detection and prediction is the use of convolutional neural networks. Related to this, we developed a new segmentation metric, wave loss. I would like to show that topological information incorporated in the loss function can be used to increase the accuracy of segmentation networks.

In light of the above, I seek to answer the following research questions:

- Can we use deep learning for high-quality, reliable noise filtering in planar bone scintigraphy?
- How robust is such a solution in real life?

- How should such a tool be evaluated where we do not have noise-free, perfect images as a basis for comparison?
- Is it possible to construct a loss function for neural network training that can take into account the topography of segmentations instead of just pixel-level comparisons?

1.2 Overview of the thesis

This section provides a brief overview of the current thesis.

Chapter 2, is an overview of the background to the topic. As the main scope of the thesis is improving the quality of bone scintigraphy, Section 2.1 reviews the imaging hardware equipment and Section 2.2 presents image mesurement protocol of bone scintigraphy. In this section I also summarized the common noise filtering methods for these kind of images. Finally, Section 2.3 introduces the basics of deep learning and segmentation methods.

Chapter 3 is the first contribution of the current thesis. It introduces a deep learning algorithm for the noise filtering of bone scintigraphy images.

In Chapter 4, Section 4.1, it is demonstrated how the lack of topological information could tamper binary object comparison, describe binary wave metric and how it can be used for shape comparison. In Section 4.2, I have introduced the extension of the wave metric to three dimensions to make it applicable on gray-scale images and two-dimensional probability distributions. Section 4.3 presents the environments and datasets we used for our measurements. Section 4.4 compares wave loss to other commonly applied metrics via commonly applied datasets and architectures.

Chapter 5 provides some insights into the possibilities for further development.

Finally, Chapter 6 provides a brief summary of the thesis.

Chapter 2

Background and theory

2.1 Overview of the imaging hardware for nuclear medicine imaging

Medical imaging devices (e.g. gamma camera, CT, SPECT, PET, MRI) are now routinely used in the clinical practice to aid diagnosis, because without them, we would not be able to see both the anatomy and the functional processes inside the body in a non-invasive way.

Until the discovery of X-rays, doctors had no available methods of obtaining an image of the inside of a patient's body. [2] The absorption of X-rays depends on the density of the material, so that a very good quality image of the bones can be obtained, while it is less suitable for examining soft tissues, due to the similar density of different soft tissues.

In 1967, Sir Godfrey Hounsfield invented the first CT scanner.[2] In this device, the X-ray source and the X-ray detector are moved around the patient together, and mathematical models are used to calculate 2-dimensional cross-sectional images using a computer. In addition to allowing soft tissue to be examined in anatomical detail, this method also provides a 3-dimensional volume by superimposing the cross-sectional images. There have been many technological advances in CT: scanning speeds have improved, resolution is constantly improving, and radiation doses are decreasing. Nowadays, CT scans can be performed in a fraction of a second, covering large areas of the body, even using more than one scan energy.

Due to the development of modern contrast agents, imaging techniques and

imaging speed, CT can also be used to monitor physiological activity, however, apart from perfusion studies, these tools are still typically used for structural imaging. [3]

Functional imaging (or physiological imaging) is a medical imaging technique used to detect or measure changes in metabolism, blood flow, regional chemical composition and absorption. In contrast to structural imaging, functional imaging focuses on the detection of physiological activities within a tissue or organ. Nuclear medicine is the most common tool for this purpose, the best known of which are various types of scintigraphy, single photon emission tomography (SPECT) and positron emission tomography. In order to complete this method of examination, a number of components and methods had to be invented. George de Hevesy is credited with the first medical use of the principle of radioactive tracing (he observed the metabolism of rats). [4]

The widespread clinical application of nuclear medicine began in the early 1950s, when knowledge about radionuclides increased. Initially, natural radioactive tracers were used, but later the development of artificial radioactive tracers was a major breakthrough. As a result, a wide range of tracers are now available to monitor many biochemical pathways and functions in the body. [4] Nuclear medicine imaging examinations are usually more organ, tissue or disease specific (e.g. lung, heart, bone, brain, tumour, infection, Parkinson's, etc.) than traditional radiological imaging which focuses on a specific part of the body (e.g. chest X-ray, abdominal/pelvic CT, head CT, etc.).

The history and development of functional imaging can be compared to the development of X-ray and CT imaging. Scintigraphy was the first to emerge, when a gamma camera is used to create 2D images of processes in the body. This technique has evolved into tomography, which gives a 3D image of the processes taking place in the body. [4]

There are two common forms of single-photon emission imaging: planar and tomographic. The planar image is a single view (projection) of the radiotracer distribution in the patient; the tomographic image is a slice or volume image of the radiotracer distribution, calculated from multiple images taken from multiple camera positions. Both imaging techniques are routinely used in nuclear medicine clinics and both use gamma cameras to collect data. Planar single-photon imaging

requires a gamma camera and a tool for displaying the acquired images; tomographic imaging requires a camera, a display method, a gantry for rotating the camera around the patient, and an image reconstruction tool. [4]

Camera-type imaging equipment has become the main instrument in modern nuclear medicine. The Anger camera invented in 1957 had become the standard for static and dynamic gamma-ray imaging tasks used for nuclear studies. Several other devices were developed as extensions of the Anger camera. Anger itself has a number of extensions and applications for the device. He used pinhole collimation to increase resolution in small regions. He created a surface projection image by rotating the patient in front of the device. camera, allowing the viewer to distinguish the lesions on the surface from the deep ones based on their different rotation speeds. Although many innovations have been made since 1958, today's clinical gamma cameras are often referred to as Anger cameras, as many of their basic features are the same as those of Anger's early designs. [4]

One of the most important components of the imaging device is a directional filtering element called a collimator. The collimator mechanically selects gamma photons travelling in a given direction for possible detection by absorbing gamma photons travelling in directions other than those specified by the collimator. [4] A parallel-hole collimator is a uniformly sized array of parallel holes surrounded by septums. The parallel-hole collimator works by transmitting all photons travelling in (or nearly in) a given direction. Ideally, gamma-photons travelling in a direction other than the direction of the holes (i.e. not perpendicular to the surface of the scintillation crystal) are absorbed. In practice, however, since the diameter of the bore is not infinitely small, there is a small range of angles of incidence that the collimator will accept. The larger the diameter of the bore, the larger the range of angles accepted and the worse the spatial resolution of the gamma camera. In reality, some photons will pass unhindered through the collimator material, while others will be scattered by the collimator material. Unwanted photons can be included in the final image, leading to image degradation. [4]

The selected gamma photons are colliding with the scintillation detector. Some of them pass through the detector without interacting with it. Those that do interact with the detector generate electronic signals that are used to estimate the location of the photon-detector interaction (spatial coordinates within the image

plane) and the energy emitted by the photon. Detected gamma-photons with energies less than the known energy of the primary emission of the radioisotope being imaged are usually rejected. The reduced energy is an indication that the photon has undergone a scattering interaction in the patient, collimator or detector and has thus deviated from its original path. Photons with reduced energy have limited information about their original point of origin and therefore their inclusion in the image reduces the quality of the image unless further processing is performed. The detection process itself consists of two steps. In the first step, the gamma-photon that survives collimation interacts with the scintillation crystal and releases energy into it. This energy is converted into more visible photons. The photons of light pass through the crystal and the light guide into an array of photomultiplier tubes (PMTs). PMTs detect photons of light; PMTs are sensitive, high-voltage devices that produce a measurable electric current from a single photon of light. [4] Each PMT outputs an electric current proportional to the number of photons detected. The light output from a scintillator is usually spatially broad and is registered by several PMTs. Specialised electronics and software are used to infer the probable gamma-photon impact point from the output of each PMT in the array. Previously, this was a simple centroid calculation performed entirely with hardware; now it is performed more accurately using statistical estimation techniques implemented with a combination of hardware and software and measured calibration data. In digital imaging technology, the image formed by a standard gamma camera image is represented on a pixel grid. The value assigned to each pixel is the number of gamma photons detected in the spatial extent of the pixel. Thus the image produced by the gamma camera is in effect a histogram of the spatial location of all the counts detected. As the number of detected unsprayed gamma-photons increases, the image noise decreases; it is therefore important to detect as many unscattered gamma-photons as possible. [4]

To fully understand the imaging process, it is important to understand the interactions of gamma photons within the patient's body and the gamma camera. The gamma photons emitted by radioisotopes in the patient may leave the patient without any change in energy or direction, or they may interact with the patient's body in one or more ways before leaving, or they may be absorbed by the body

completely. Three types of interactions affect imaging: photoelectric absorption, Compton scattering and coherent scattering. Photoelectric absorption, which is sometimes accompanied by the emission of low-energy fluorescent gamma rays that typically do not escape from the patient, is the complete absorption of a photon by an atom. Photoelectric absorption in the collimator and crystal of a gamma camera is a common interaction of gamma photons that exit an object and interact with the gamma camera during imaging. Absorption of a photon in the collimator means it is not detectable; absorption of a photon in the scintillation crystal means it is detectable. Compton scattering of photons by electrons changes the direction and energy of the photons. Coherent scattering, on the other hand, typically results in a smaller change in propagation direction and an insignificant change in energy. At the gamma-emission energies of radionuclides used in diagnostic nuclear medicine, coherent scattering in tissues accounts for less than a few percent of the total scattered photons. It is not easy to characterise the path of the scattered photon and it is impossible to determine precisely its initial propagation direction. Most scattering events reduce the energy of the photon, which can be used to exclude unwanted contributions of such photons to the image. On the other hand, the energy resolution of scintillation detectors is not sufficient to distinguish all scattered photons from non-scattered photons. As a result, Compton scattered photons as well as coherently scattered photons are often included in the images and cause a loss in image contrast. (Gamma photons propagating in specific directions pass through the gamma camera collimator. Most, if not most, of these are absorbed or detected by the camera's scintillation detector. However, not all the detected photons are included in the images. Some detected photons are rejected because their energy does not fall within the specified energy ranges). [4]

2.2 Overview of bone scintigraphy

In my PhD thesis, I worked on improving bone scintigraphy, which is a specialised radiological technique for examining different bones in the skeleton. It is used to identify abnormal changes of bone tissues. Bone scans can also be used to monitor the progress of treatment of certain diseases.

The diagnostic procedure of bone scintigraphy involves the intravenous injection of Tc-99m-labeled diphosphonate molecules (e.g. MDP, HMDP) into the patient. The source distribution in the body is recorded with a gamma camera in a so-called whole-body scan. The signal-to-noise ratio of the resulting images can be attributed to several different factors. The most important of these are the amount of activity injected, the duration of the measurement, the time of the accumulation of the radiopharmaceutical, the degree of radiopharmaceutical coupling and the sensitivity and other properties of the gamma camera hardware. Also, the absorption of photons by the patient's body, which greatly reduces the signal-to-noise ratio, should not be neglected [4].

Whole-body bone scans are initiated by intravenous injection of 99mTc-MDP (methylene diphosphonate) or a similar compound, and imaging begins 2-5 hours after injection. Uniform bone uptake usually indicates a normal examination. Focal uptake (local uptake that differs in magnitude from uptake of adjacent bone) may indicate abnormality. If the focal uptake is larger than that of adjacent bones, it may indicate arthritis, fracture or metastasis. A focal uptake smaller than that of adjacent bones may indicate a necrotic tumour, a lytic lesion or the result of radiotherapy. [4] No commercially available gamma camera is large enough to take an image of an entire average-sized adult without moving the camera or the patient. Therefore, whole-body bone scans are performed by moving the camera along the longitudinal axis of the patient or, correspondingly, by moving the patient longitudinally alongside the camera. To perform a bone scan the cameras are positioned at 90° and 270° with the patient lying supine on the table with the legs turned towards the gantry. The table and the patient are then moved to the starting position of the acquisition, which places the patient's head in the FOV of the cameras. During the scan, the patient and the table are moved to obtain an image from the patient's head to the feet, in that order. Note that to obtain whole-body images that accurately represent the relative values of the radiotracer images of the head, chest, abdomen and legs, the acquisition and motion parameters must be accurately coded and matched. Most whole-body bone scans are performed using a system with two gamma cameras, so that front and rear views can be recorded simultaneously.

However, the identification of pathological lesions and accumulations requires considerable medical and radiological work as these devices in some cases result in large number of images, which often represent ambiguous information. This is especially true for tomography-type scans, where after the reconstruction the output is a whole 3D volume, consisting a lot of slices, which can be viewed from different angles, so the analysis is an exhaustive process. For this reason, machine learning (ML) based image processing techniques are becoming more and more widespread nowadays, which are able to filter, highlight, segment or classify abnormal lesions in the image. The characteristic of these ML-based solutions is that the character (feature vector) of the lesions is not determined in a "hand-designed manner", but is identified and selected by a self-learning process, e.g. such auto-representation is generated in deep layers of convolutional neural networks (CNNs). [5] [6]

A peculiarity of ML algorithms is that they typically require a vast amount of annotated medical images to operate with sufficient accuracy. For example, in order for a benign/malignant lesion classification algorithm to work with the highest possible sensitivity (true positive rate) and the highest possible specificity (lowest possible false positive rate), it is necessary to obtain a sufficiently large number and various type of benign and malignant lesions, that is, the broadest possible spectrum of lesions should be included in the training database, and preferably in a balanced distribution. Although a huge number of medical images are produced nowadays, obtaining an image database is a difficult task, mainly due to ethical, property and privacy restrictions [7]. This is a serious impediment to the development of deep learning based algorithms, which typically need to be trained on a few 100 or in a more complex case a few 1000 images and ensure adequate performance in clinical settings [8].

An important component of computer aided diagnosis (CAD) algorithms is noise filtering, as these medical images typically have a high noise content. Improving the signal-to-noise ratio makes it possible to obtain an image of sufficient quality with considerably less administered activity (e.g. SPECT, PET) or X-ray dose (CT), which reduces the radiation exposure of both the physician and the patient and thus the risk of the examination. A particularly high signal-to-noise ratio can be achieved by using noise-filtering neural networks (NNs), which

can be configured and trained to act as specialised edge preserving noise filters. The training image database implicitly contains both image structures and a characteristic noise spectrum, from which the neural network can synthesize a filtered, noise-free image [9].

An important characteristic of modern NN-based noise filters is that, in addition to the noisy data used as input to the network, the reference ("ground truth") images also contain considerable noise during the training phase. Nevertheless, it has been demonstrated that a properly constructed NN filter trained in this way can synthesize filtered images with a better signal-to-noise ratio than the reference images and outperforms conventional noise filters, e.g. BM3D method, for either Gaussian or Poisson noise [1]. Considering that the image database used for filter training usually consists of a highly limited number of images, we consider it especially important to investigate the robustness of the trained NN-based image processing algorithm, i.e. its sensitivity to the noise content and distribution of the images according to different aspects, including patient age, gender, body mass index value (BMI), and the nature and distribution of characteristic pathological structures in the images. This analysis can reveal the robustness of such an image processing algorithm, either on its own or as part of a larger CAD system deployed in clinics around the world.

The development of robust noise filters is a particularly important task because the contrast and visibility of lesions is highly dependent on the signal-to-noise ratio of the image. However, this task is a huge challenge because noise suppression and contrast preservation or enhancement usually work against each other.

Long measurement times are an everlasting problem in diagnostics and by enabling shorter measurements and maintaining the same diagnostic quality, we can perform patient measurements more efficiently, providing greater throughput with a given device. And as well as allowing more patients to be diagnosed, the patient's comfort is also improved as they have to lie still for a shorter period of time. In addition, the risk of movement would be reduced, which would mean better images overall, as fewer scans would need to be repeated. However, in terms of image quality, the injected activity also plays an important role. By reducing the administered activity, the radiation exposure to the patient and assistant is reduced and so the risk of the procedure can be significantly reduced.

2.2.1 Mathematical formulation of noise filtering

To formulate noise filtering mathematically, we introduce the following notations. Let $\boldsymbol{x}$ be the input data with significant noise content, which may result for example from lower administered activity or shorter measurement time compared to the normal recording protocol, and let $\boldsymbol{y}$ be the recording made with the normal protocol. For noise filtering in this case, our goal is to find a mapping $f()$ where:

$$y \approx f(\boldsymbol{x}) \tag{2.1}$$

The f used to cover traditional noise filtering algorithms such as Gaussian or median filtering, or possibly more complicated methods such as BM3D [10]. In this thesis, we have trained a neural network for the role of f, which can synthesize filtered images with a better signal-to-noise ratio and better diagnostic image quality than the previous ones.

2.2.2 Traditional noise filtering solutions for bone scintigraphy

In addition to the commonly used Gaussian and median filtering in the literature [11], currently Block-matching and 3D filtering (BM3D) algorithm is considered as one of the best noise-filtering algorithm. The BM3D filter works by dividing the image into patches of equal size, finding the patches that are most similar to each reference patch, and then filtering them in a grid over the resulting 3D domain. The construction of the 3D domain itself is called block matching.

It then considers that the similar patches are correlated and the noise sitting on them can be removed by decorrelation. The reference patch is slid through the entire image pixel by pixel, performing the above operation at each step, and thus obtaining a denoised patch around each pixel. Then, the overlapping patches are added together with weights decreasing by the distance from the center pixel to obtain the final filtered image. [10]

In contrast to noise-filtering solutions based on neural networks, in order to use the BM3D filter with adequate performance, an Anscombe transformation had to be performed on the input data. This is a variance stabilizing method that can transform a probability variable with a Poisson distribution into a variable with an

approximately standard Gaussian distribution. [12] The BM3D implementation used in my thesis is available at [13].

The parameters of the algorithm were set based on grid-search optimization, optimizing for the highest possible score according to our evaluation method presented in my thesis. Instead of pure Anscombe transformation, we achieved better image quality by using the generalized Anscombe transformation [14], see eq. (2.2). After the BM3D filtering, the final result is inverted back to the original domain using the closed-form approximation of this exact unbiased inverse [14], see equations (2.3) and (2.4).

$$A_G(\overset{*}{z}) = \begin{cases} \frac{2}{\alpha}\sqrt{\alpha \overset{*}{z} + \frac{3}{8}\alpha^2 + \overset{*}{\sigma}^2} - \alpha\mu, & \overset{*}{z} > -\frac{3}{8}\alpha - \frac{\overset{*}{\sigma}^2}{\alpha} + \mu \\ 0, & \overset{*}{z} \leq -\frac{3}{8}\alpha - \frac{\overset{*}{\sigma}^2}{\alpha} + \mu \end{cases} \tag{2.2}$$

Closed form approximation of unbiased inverse Anscombe transformation [14]:

$$A^{-1}(x) = \frac{1}{4}x^2 + \frac{1}{4}\sqrt{\frac{3}{2}}x^{-1} - \frac{11}{8}x^{-2} + \frac{5}{8}\sqrt{\frac{3}{2}}x^{-3} - \frac{1}{8} - \sigma^2 \tag{2.3}$$

Closed form approximation of unbiased generalized inverse Anscombe transformation [14], where A^{-1} comes from eq. (2.3):

$$A^{-1} : x \mapsto \begin{cases} A^{-1}(x)\alpha + \mu, & A^{-1}(x) > 0 \\ \mu, & A^{-1}(x) < 0 \end{cases} \tag{2.4}$$

In eqs. (2.2), (2.3) and (2.4) $\overset{*}{z}$ is the observed pixel value obtained through an image acquisition device. In Reference [14] they model each $\overset{*}{z}$ as an independent random Poisson variable p with and underlying mean value y, scaled by $\alpha > 0$ and corrupted by additive Gaussian noise $\overset{*}{n}$ of mean μ and standard deviation $\overset{*}{\sigma}$.

2.3 Deep learning

Machine learning (ML), a class of artificial intelligence techniques in which a computer captures patterns in data. The learning takes place without explicit programming and uses the learned patterns, features to support decision-making. [15] A special type of machine learning is deep learning (DL). DL is a class of machine learning that automatically learns hierarchical features of data using

multiple, large amount of layers. [16] The successful application of this method overwhelmingly defeated previous ML methods for visual recognition tasks in a 2012 competition called ImageNet [[17], [18]]. DL has since gained acceptance, succes and popularity in a variety of scientific and industrial domains, including not only computer vision, but also speech recognition, drug discovery and bioinformatics [19], [20], [21]].

DL typically means deep neural networks. Thus neural networks, a type of machine learning, are the basis of recent deep learning techniques.

The perceptron is the earliest neural network model consisting of a single layer. [15] The inputs to the model can be different features (e.g.: features of a lesion, such as tumor size, intensity). The parameters of the model, W = (w1, w2,...,wn), are multiplied by the inputs and then the activation function is applied. An example of a simple activation function is that the output is 1 if the value is greater than 0, otherwise the output is 0. During training, the parameters of the model are optimized for proper decision making. Training can be formulated as minimizing the error between the output of the model and the expected outcome. In practice, the gradient descent method is used for training, which iteratively updates the model parameters according to the gradient of the error function [22].

The single-layer perceptron has limitations for complex, nonlinear data patterns. To overcome this, hidden layers have been added between inputs and outputs. This type of neural network is a well-known traditional neural network, the multilayer perceptron. Although the use of hidden layer has improved the performance of neural networks, training of these deep architectures is difficult. Deep neural networks gained attention after the use of a new training method, pretraining [23], [24]. While pretraining initiated the popularity of deep learning, current models, especially in the image analysis domains, have shifted towards specialized architectures, for example, in the field of image processing, convolutional neural networks (CNN) have become predominant. These specialised architectures are designed to incorporate and apply a wealth of knowledge that is specific to each domain. Today's neural networks therefore incorporate specific engineering solutions to help solve particular problems.

2.3.1 Convolutional neural networks

CNN usually uses pixels, voxels, directly as input (instead of preprocessed features). To use the structural information of adjacent pixels, convolution layers are used instead of densely connected perceptron layers. A convolution layer consists of a fixed number of convolution filters. For convolutional layers, the values of the convolutional filters are the weights (parameters), that can be learned and are optimized during the training process. Usually, an activation layer follows the convolution layer as in the case of perceptron. After the convolution layers, pooling layers are usually added to subsample the feature maps to aggregate the information. A dramatic improvement in image recognition was achieved in 2012 by AlexNet, which consists of five convolutional layers [18]. In 2014, the winner of ImageNet was a 22-layer network called Inception [25], and in 2015, a 152-layer network called ResNet further improved the performance of image recognition [26]. Since then, countless new architectures and developments have been released every year.

2.3.2 Training neural networks for segmentation tasks

The solution of segmentation problems with deep neural networks requires a well-defined loss function for comparison and network training. In most network training approaches, only area-based differences that are of differing pixel matter are considered; the distribution is not. Our brain can compare complex objects with ease and considers both pixel level and topological differences simultaneously and comparison between objects requires a properly defined metric that determines similarity between them considering changes both in shape and values. In past years, topographic aspects were incorporated in loss functions where either boundary pixels or the ratio of the areas were employed in difference calculation. In this thesis I will show how the application of a topographic metric, called wave loss, can be applied in neural network training and increase the accuracy of traditional segmentation algorithms. Our method has increased segmentation accuracy by 3% on both the Cityscapes and Ms-Coco datasets, using various network architectures.

The application of neural networks and modern machine-learning techniques opened up various applications for image segmentation, where instead of or additionally to bounding box detection a pixel level segmentation of input images can be created. In recent years, segmentation networks have become ubiquitous in computer vision applications, since they usually provide better understanding of scenes than classification or detection with bounding boxes.

These methods are applied in various tasks, from medical imaging [27] to self-driving cars [28]. These methods may vary depending on the selected architectures (U-Net [27], SegNet [29], Mask R-CNN [30], RetinaNet [31]) or even on the exact specification of the segmentation problem (semantic segmentation [32], instance segmentation [33] or amodal segmentation [34]), but all of these approaches require a metric which will compare the actual network output to the expected, ideal outcome or ground truth.

These distances are indispensable for classification, data clusterization or in the application of any modern supervised learning method for artificial intelligence. From an engineering point of view, a metric is inherently a simplification of the problem representation, which condenses similarity or difference between two high-dimensional data points into a scalar value and if significant and important data is lost during this projection the algorithm cannot provide correct results.

Apart from information compression, one may have another important expectation about a proper metric, which works against the generality of information compression. On the one hand, a metric has to be sensitive enough to allow comparison in an abstract space; meanwhile, on the other hand, it has to be robust to filter out noise.

In current applications, in almost all cases a pixel-based distance is applied, where two images are compared to each other according to a given metric (like L1, L2 or Smooth-L1 [35] distances). Similarly the outcome image and the ground truth can be considered as probability distributions and cross entropy can be applied to determine a distance between them, but none of these metrics take into account the position of the differences.

It is not our aim to speak against intensity-based distances and loss functions, but we would like to demonstrate that a metric involving topological information

about the shape of the object and relative position of the differences can have additional value in network training.

The representation of topological information in loss calculation has appeared in the past year in various papers, such as [36, 37, 38], but all these approaches at their core calculate the pixel-wise differences and approximate topology using persistence barcode calculation [37] or skeletonization [38].

One can easily see that in the case of a perfect solution the loss will indeed be zero for every metric and higher losses will encode either a larger area of altered pixels, larger intensity differences or both. However, in the case of errors with similar values, the position and shape of the misclassified pixels will also matter. For example, a hundred differing pixels can be organized into an arbitrary shape such as a circle, a line or randomly placed separated points and the position of these differences should also determine the quality of a solution in the case of segmentation.

Additionally, loss functions should also identify those regions which are responsible for the error. In the case of segmentation, falsely detected pixels around the real object and not segmented pixels inside a homogeneous region are usually caused by the false detection of the boundary and not because of the exact pixels at that position. This weighting is implicitly present in the network via the downscaling operations in different layers, but it is advantageous to explicitly find the source of the error at the calculation of the loss function. These aspects are present in boundary losses, such as in [39], where the boundary regions of the ground truth masks are handled with increased importance, but the topology of other regions is not represented at all.

Chapter 3

Proposed deep learning based noise filtering

In deep learning, one of the most important steps is the compilation of the training data set and feature engineering, which determines what information will be the input to the neural networks and in what format. In order to ideally prepare the data and select the set of augmentations that can be used, it is necessary to take into account the characteristics associated with the bone scintigraphy imaging method.

Our measurement statistics follow a Poisson distribution and the noise and signal in the gamma camera images are comparable in magnitude. Our goal in noise filtering is to estimate the expected value of the signal based on the values measured by the detector.

Autoencoder-based convolutional neural networks are particularly suitable for noise filtering solutions. [40]. A common solution for this type of method is to generate the input image by degrading the noise-free image and preserving the noise-free version as expected output. A recent study has shown that neural networks can be trained without noise-free images, using a method called Noise2Noise [1]. When we put the same recording on both sides of a neural network, we risk that our network will fall into learning the identity function, so it does not do any useful work for us. However, we can guarantee that the two sides are different by using an augmentation technique. We put an image with high noise content (with artificially reduced quality) on the input and a less noisy version of it on the output. If we generate this noise in a properly randomized way, we end up

with a very powerful augmentation technique, whereby we change not only the input side, but also the output side.[41]

According to the article [42], one possible strategy is to put an artificially degraded, noisy image on the input side and keep the original image on the expected side. The authors found that putting (statistically independent) recordings with half statistics on opposite sides gives a more accurate result, at least in the sense that it better approximates the hypothetical case where we have a noise-free, perfect recording. An image with halved statistics means that it has been degraded as if it had been measured for half as long. For planar bone scintigraphy, this phenomenon is more pronounced, since we have a much noisier image on normal statistics than on normal CT. The planar bone scintigraphy image is very similar to a CT measurement, which can be modelled by the sum of a Poisson distribution and a Gaussian distribution [43].

According to Reference [44], if we neglect the Gaussian component, we can use binomial sampling to split the record into two independent records with worse statistics. Let Z be an $\mathbb{N}_0$-valued random variable and let Z_1, $Z_2 \ldots$ be a sequence of independent random variables that have a Bernoulli distribution with parameter $p \in [0, 1]$. If Z and $(Z_n)_{n>=1}$ are independent, then the random variable

$$X := \sum_{j=1}^{Z} Z_j \tag{3.1}$$

is called a p-thinning of Z, where we set $X := 0$ if $Z = 0$. This means that the conditional distribution of X given $Z = n$ is binomial with parameters n and p. Let $p \in [0, 1]$. Let Z have a Poisson distribution with parameter $\gamma > 0$ and let X be a p-thinning of Z. Then X and $Z - X$ are independent and Poisson distributed with parameters $p\gamma$ and $(1 - p)\gamma$, respectively.

3.1 Training strategy

From a single recording, one can produce countless different recordings with different statistics, or for a given statistic, an infinite versions of it. The question arises as to whether it is worth training separate neural networks for each statistic, or whether it is sufficient or more appropriate to train a general network.

For planar bone scintigraphy, it can be said that since many factors influence the measured image quality, it is difficult to estimate the image statistics that is the actual noise content of the image. It may vary depending on the amount of radiopharmaceutical administered, the waiting time, the factors determining the enrichment in the patient's body or the time of imaging. Thus, it is difficult to determine the statistics of an image, so it is more useful to train a general neural network that can handle any statistics.

This was implemented as follows:

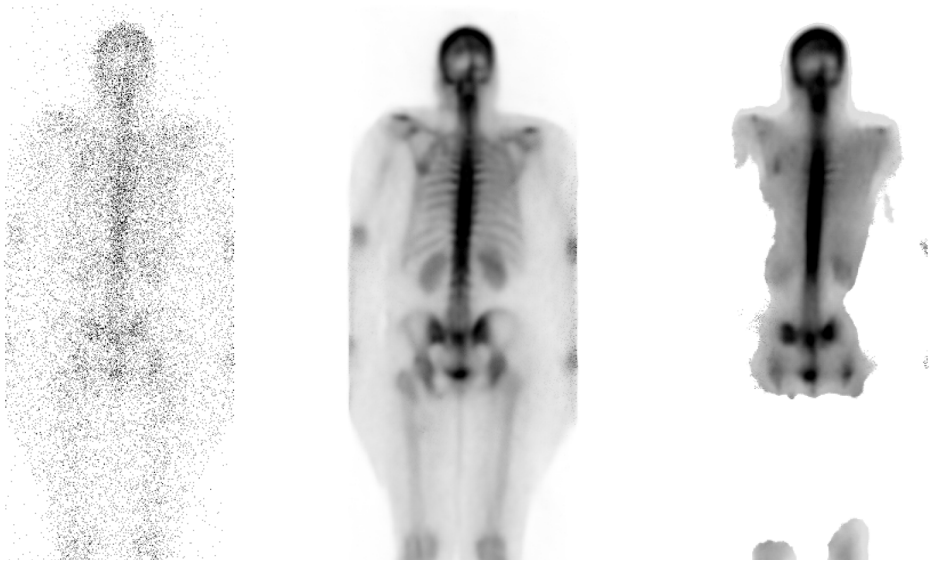
All recordings were resampled based on a binomial distribution, artificially generating realistically degraded recordings as if they had been taken at one-third, one-quarter, one-eighth, etc. recording times.

If we subtract this degraded image from the original measurement, we get an independent record with better statistics than the data on the input side. The task of the neural network is to estimate the transformation between the two generated recordings. To avoid the network having to learn a multiplicative factor between the input and the expected side, we scale (normalize) the training data on the expected side.

We used Mean Absolute Error (MAE) as a function of the learning loss between the actual filtered and reference image, as was done in the [42] article. Although there are advantages to using the Mean Squared Error (MSE) error function, because compared to MAE no local background erasing occurred at extremely low statistics (Fig. 3.1), but it was less effective at filtering noise in several regions of the image (Fig. 3.2). In our final model, we used MAE as error function and used hyperparameter optimization (rate of Poisson thinning, learning rate) to eliminate problems in low statistics regions. However, only for localisation purposes (taking a quick measurement to plan a long measurement), at very high noise levels, we recommend using the MSE loss function.

We used the commonly applied methodology: the backpropagation algorithm and gradient based optimization to train the neural networks [45].

Our trained neural network acts as a noise filter, producing a 2D noise-filtered image. The input is a low-quality, noisy measurement, and the output is a smoothed, contrasty image, similar to what would be expected if the input had been recorded over a very long time with good statistics.



(a) input (b) MSE (c) MAE

Figure 3.1: With greatly reduced statistics (1/32) compared to normal, the network trained with the MAE loss function tends to cut out or zero out low-impact regions during filtering. The figure shows a very strong version of this phenomenon.

3.2 Neural network architecture

The authors of Ref. [46] have shown that these autoencoder-based systems are particularly robust for feature extraction even in the presence of severe noise. In the field of image processing, meanwhile, convolutional neural networks have become extremely widespread. These networks have been used mainly for classification, and their rapid spread and reputation is due to AlexNet [47], which has achieved outstanding results on ImageNet.

For the sake of simplicity, this thesis is limited to introducing a solution based on the famous U-NET architecture. [27]

Autoencoder-type networks are among the so-called image-to-image transformation networks. The U-NET type, an evolution of these networks, was a major advance in segmentation, but was later successfully applied in other areas. [27]

This artificial neural network combines different layers of convolution [45] and max-pooling [45] see Figure 3.3. The peculiarity of this network lies in its skip-connections, whereby lower layers receive textural information from higher

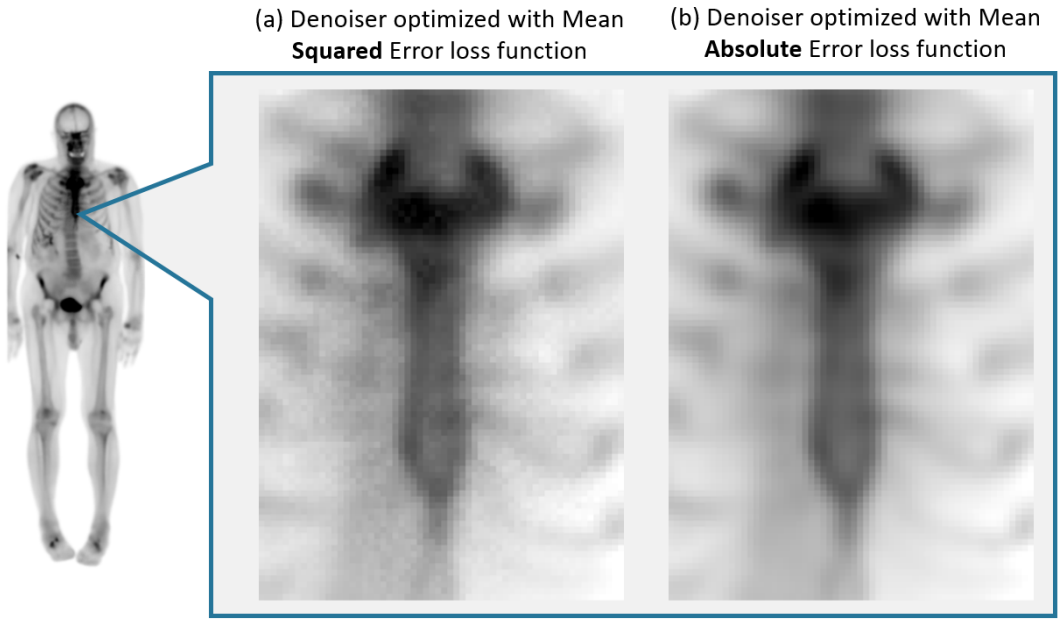


Figure 3.2: Choosing MSE as the optimization cost function, (a) the network tended to be less effective at filtering, while the usage of MAE (b) results in a smoother image.

layers. The skip connections also facilitate the propagation of the gradient. The resource requirements (GPU memory, computational need) of the network are relatively low compared to its complexity. [27]

3.3 Materials and methods

3.3.1 Characteristics of the clinical data used

The selection of data suitable for training the neural networks was done with the Q-Bot software. [48] For the development, 2430 anonymized recordings (from 1215 patients, anterior and posterior) were used, acquired by AnyScan® DUO/TRIO SPECT/CT (Mediso Ltd.) and InterView^{TM} processing SW (Mediso Ltd.). All patients were given 5-600MBq Tc-99m methylene diphosphonate (MDP) (Isotope Institute LTD, Budapest, Hungary) intravenously with 2-5 hour accumulation time. The matrix size was 256*1024 with 130mm/min scanning speed. No additional filters were used.

From the 2430 measurements we used 1886 acquisitions as training data for optimization of the network's weights and we set aside 544 for evaluation purposes.

3.3.2 Detailed description of neural network architectures

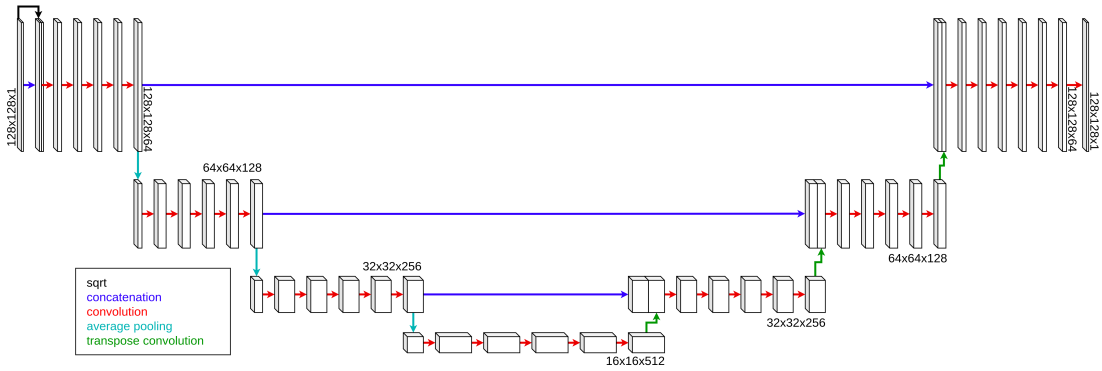


Figure 3.3: The network architectures used for noise filtering were all U-NET based. The differences between the architectures were in the number of convolutional layers per level, and the number of convolutional filters. The network, named *L-NN*, contained 4 levels and always had 5 convolutional layers following each other. The filter numbers of the convolutional blocks used at each level were 64, 128, 256 and 512. The neural network named *S-NN* also contained 4 levels, always with 3 convolutional layers following each other and the number of filters used at each level were 16, 32, 64 and 128.

In this thesis, we present the results of two neural networks with U-NET architecture [27], referred to as *S-NN* and *L-NN*.

Both neural networks start by concatenating the root of the input to the input and all of the activation functions were ReLU (Eq. (3.2)) in the networks [49].

$$\mathrm{ReLU}(x) = max(0, x), x \in (R) \tag{3.2}$$

The network builds from an encoder branch, which compresses the information, and a decoder branch, which reconstructs the image. The encoder branch has encoding blocks consisting of convolutional layers and an average pooling layer. These convolutional layers contain a set of filters (kernels), which are learned through the training process. Each kernel convolves with the image and creates a feature map which will serve as input for the next layer. [49] After each level on

Layer		Feature size	Channels		Kernel size	Layer Repetition		Block Repetition
			S-NN	L-NN		S-NN	L-NN	
Sqrt		128×128	2	2	-	-		-
Encoding block	Convolution	$\frac{128}{2^{n-1}} \times \frac{128}{2^{n-1}}$	$16*2^{n-1}$	$64*2^{n-1}$	3×3	3	5	$n = 1, 2, 3$
	Average Pooling	$\frac{128}{2^n} \times \frac{128}{2^n}$	$16*2^{n-1}$	$64*2^{n-1}$	3×3	-		
Convolution		16×16	128	512	3×3	3	5	-
Decoding block	Transpose Convolution	$\frac{128}{2^{n-1}} \times \frac{128}{2^{n-1}}$	$16*2^{n-1}$	$64*2^{n-1}$	2×2	-		$n = 3, 2, 1$
	Concatenation	$\frac{128}{2^{n-1}} \times \frac{128}{2^{n-1}}$	$16 * 2^n$	$64 * 2^n$	-	-		
	Convolution	$\frac{128}{2^{n-1}} \times \frac{128}{2^{n-1}}$	$16*2^{n-1}$	$64*2^{n-1}$	3×3	3	5	
Convolution		128×128	1	1	1×1	-		-

Table 3.1: Details of the neural network architecture. The differences between the architectures were in the number of convolutional layers per level, and the number of convolutional filters. Both neural networks contained 3 downscaling operations, so they were U-NET architecture networks with 4 levels.

the encoder branch, we decrease the resolution of the feature map with average pooling by a factor of 2.

The decoder branch has decoder blocks, which consist of transpose convolution layers [27], concatenation layer, and convolutional layers. In these blocks, the transpose convolution layers double the resolution, the concatenation layer concatenates the results of the corresponding encoding blocks before the convolution.

The network, named *L-NN*, contains 4 levels and always had 5 convolutional layers following each other. The filter numbers of the convolutional blocks used at each level were 64, 128, 256 and 512. The neural network named *S-NN* also contained 4 levels, always with 3 convolutional layers following each other and the number of filters used at each level were 16, 32, 64 and 128.

The reason for including two networks of different sizes in the thesis is to show that our learning strategy is stable and does not depend heavily on the exact network architecture and the fine-tuning of its hyperparameters. The differences

between S-NN and L-NN are only the number of convolutional layers and the number of filters in them. S-NN is a smaller network, which requires fewer computing resources and generally less prone to over-fitting than larger neural networks [49].

A detailed summary of the architectures can be found in Table 3.1.

For training, we used NVIDIA GTX 1080 GPUs and relied on Keras [50] as ML software, with TensorFlow 2.4.1 backend library. [51]

As optimizer we chose Adam [52] optimizer and we trained the networks with a learning rate of 0.001 for 1200 epochs. We did the training on 128x128 cropped patches instead of using the whole images for larger possible batch size [53], and also considered this method as an augmentation. In practical use, we run the neural network on the whole image during prediction, taking advantage of the fact that it is a fully convolutional network, so the network is invariant to the size of the input image.

It is important to note that if the same method is used to produce the data in the validation set as in the case of the training data set, then we cannot determine whether the current state of the network is actually better than the previous state based on the loss function alone. We can see the evolution of the training loss (MAE) as a function of the training epoch in Fig. 3.4 . The main reason is that both our input and expected data are noisy, so we can easily choose a suboptimal network as the best model. What is worth doing, however, is that we do not apply augmentation on the validation dataset, i.e. we evaluate the network on the same set of images at each epoch.

The evaluation method used in the evaluation section of this thesis provides a solution to qualify the trained networks.

The process of choosing the best model is the following:

- Training of multiple neural networks with our training strategy based on binomial sampling
- Best model selection based on validation loss
- Creation of the evaluation framework
- Evaluation of the models with the framework

Once the evaluation framework is established, it is recommended to use the scores of our evaluation method instead of calculating the validation loss when training new neural networks.

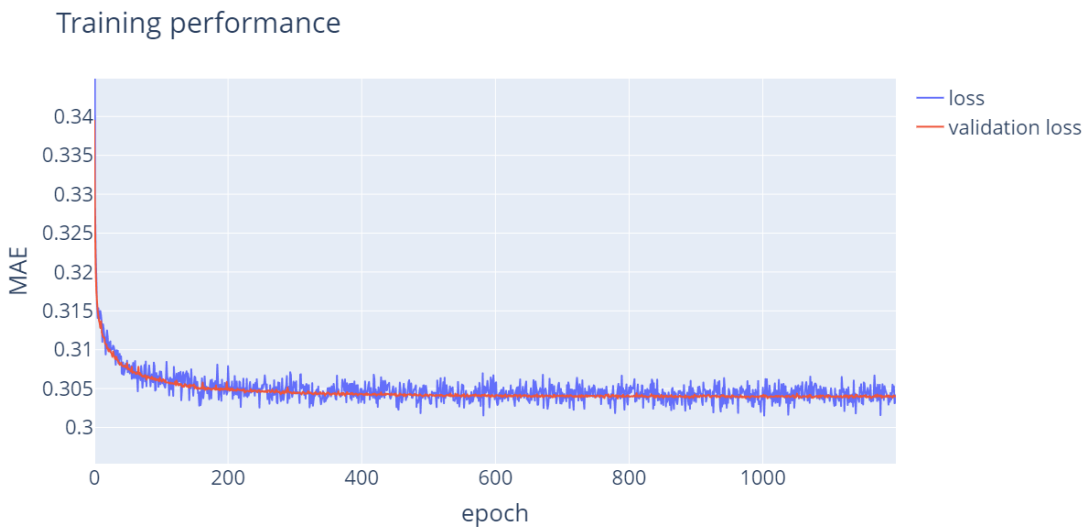


Figure 3.4: The training performance of a neural network called S-NN. It can be seen that the error function is noisy due to the augmentation of the training data. Our validation data is static, where we can observe a slow improvement in performance.

3.4 Evaluation and results

A common problem with emission imaging is that we do not have a true, noise-free, high-quality image of a patient to use as a reference. This is usually avoided by measuring physical phantoms, in which the amount and distribution of activity loaded is known, so that the quality of the image produced by imaging can be determined. However, for deep learning based noise filtering solutions, we cannot use phantoms. On the one hand, if the solution is trained on patient data, the performance of the neural network is suboptimal in the presence of phantoms. On the other hand, if the neural networks have been trained on phantoms, we can easily over-train them.. In other words, we can obtain a filter that performs outstandingly on phantom measurements. However, the variability of physical phantoms is not

significant enough, and the deceptively good performance measured in this way would not give us useful information about the real-life performance of the system.

Therefore, the following solution was used as an evaluation method: At a late stage of the development, we selected a neural network that we judged to have sufficiently good performance on measurements with low noise content. With this neural network, we created noise-filtered images from the evaluation dataset (544 measurements), which were then examined by physicians to see if there was any unusual structure, accumulation or artifact in the image, compared to the original unfiltered image. In our evaluation process, we considered these images as noise-free, expected *ideal* images, from which we generated images with normal statistics using Poisson noise. These normal statistics images were used as input to our solutions and we also used them to produce lower quality images by binomial sampling. The whole pipeline and the examples of the images produced by the pipeline are shown in Figure 3.5 and Figure 3.6. Using this process, we can also measure the peak signal-to-noise ratio (PSNR) [54] of the filtered images, which is shown in the figure examples. Given a reference image y and a test image x, both of size MxN, PSNR in 2D is defined as Eq. (3.3).

$$\mathrm{PSNR}(x,y) = 10\log_{10}\frac{(\max_{j=1}^{n} y_j)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{3.3}$$

The filtered images were compared to the *ideal* images using Root Mean Square Error (RMSE) Eq. (3.4), and Structural Similarity Index Measure (SSIM) [55] Eq. (3.6) and MAE Eq. (3.5) metrics. The value was calculated only for those pixels where the intensity was greater than zero in the *ideal* image. This was necessary because the different amount of foreground-to-background ratio in the scintigraphy images distorted the per-image metrics.

$$\mathrm{RMSE}(x,y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{3.4}$$

$$\mathrm{MAE}(x,y) = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{3.5}$$

$$\mathrm{SSIM}(x, y) = \frac{1}{n} \sum_{i=1}^{n} \frac{(2\mu_{x_i}\mu_{y_i} + C_1) + (2\sigma_{x_i y_i} + C_2)}{(\mu_{x_i}^2 + \mu_{y_i}^2 + C_1)(\sigma_{x_i}^2 + \sigma_{y_i}^2 + C_2)} \tag{3.6}$$

In Eq. (3.6) μ_{x_i} and μ_{y_i} are the local mean intensity of x_i and y_i respectively. Let R be the data range of the image (distance between minimum and maximum possible values), then $C_1 = (K_1 R)^2$ and $C_1 = (K_2 R)^2$, where K_1 and K_2 are constants. $\sigma_{x_i y_i}$ is the correlation of x_i and y_i; σ_{x_i}, σ_{y_i} are the local standard deviation at x_i and y_i.

For calculating the SSIM index we used *skimage.metrics.structural_similarity* function from the scikit-image library (version 0.17.2) [56]. We used the default parameters of *skimage.metrics.structural_similarity*, so we used 7x7 window size for calculating local mean intensity, $K_1 = 0.01$ and $K_2 = 0.03$ constants.

The maximum intensity of the images was saturated at 255 because we did not want the differences in the injection point, bladder and other high-intensity areas - irrelevant for diagnostics - to overly determine the judgment of the performance of the filters.

3.4.1 Comparing the performance of different neural networks and conventional noise-filtering solutions

Methods based on neural networks were first compared with conventional noise-filtering solutions for normal, 1/3 and 1/9, 1/16, 1/32 statistics.

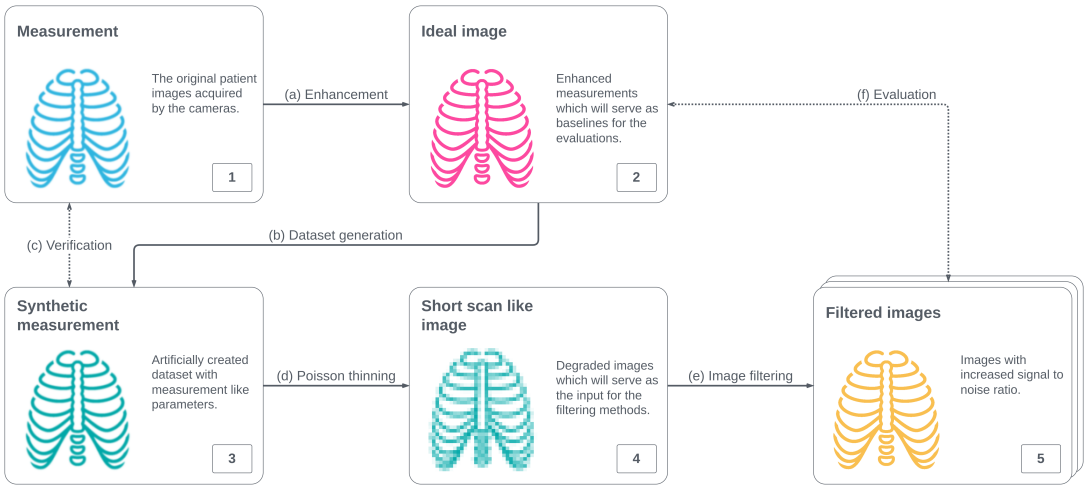
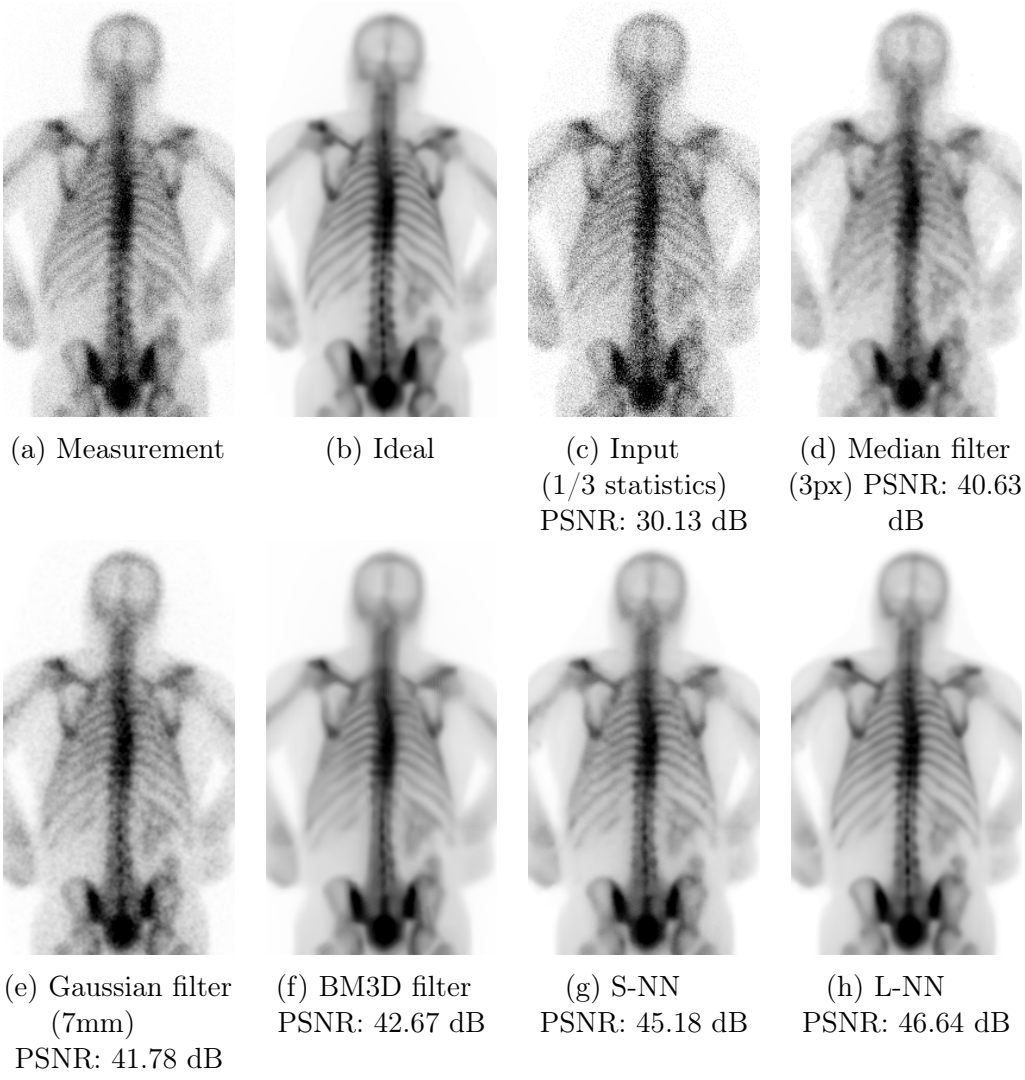


Figure 3.5: Evaluation pipeline: We start from the real measurements acquired by the scanner (1). The second step is to create a noise-free image (2) with a reference enhancement solution (a), which was a neural network based denoiser in our case [41]. The ideal image will be then examined by physicians to see if there was any unusual structure, accumulation or artifact in the image. From this noiseless ideal image we generate synthetic measurement (3) with adding poisson noise (b), which will be verified (c) by statistical tests. The next step is to construct the records with worse statistics (4) using Poisson thinning (d). Finally these images will be the inputs to the various filtering tools (e), which results' (5) will be compared (f) to the ideal images (2).

The BM3D implementation used in my thesis is available at [13]. We obtained the parameters by running hyperparameter optimizations on a few selected images to get the best results according to our evaluation method, and then we checked the results by eye. Due to the complexity of the BM3D algorithm, many parameters can be configured. The authors of Reference [13] have created several well-configured configurations, from which we have selected the *vn_old* profile using grid search based optimization. Although the *'vn'* profile was proposed [57] as a better alternative to the profile originally proposed in [58] (which is currently the



(a) Measurement (b) Ideal (c) Input (1/3 statistics) PSNR: 30.13 dB (d) Median filter (3px) PSNR: 40.63 dB

(e) Gaussian filter (7mm) PSNR: 41.78 dB (f) BM3D filter PSNR: 42.67 dB (g) S-NN PSNR: 45.18 dB (h) L-NN PSNR: 46.64 dB

Figure 3.6: Evaluation pipeline: We start from Measurement (a), from which we create a noise-free image with a reference filter (b). This will be then reviewed by doctors and taken as a benchmark. From this we generate an artificial degraded noisy image (c). Images (d), (e), (f), (g), (h) show the results of different filters. Since we have the noise-free reference image, we can correctly compute the errors of each method using the metrics.

vn_old profile in the library), we still got better results with the previous preset. For the Generalized Anscombe transformation parameters (see Eq. (2.2), (2.3) and (2.4)) we used $\mu = 0$, $\sigma = 8$ and $\alpha = 1$ and for BM3D the *sigma_psd* (which is the noise power spectral density) 0.8 proved to be the best.

In addition to the BM3D filter, we have included 6 different Gaussian filters with full widths at half maximum (FWHM) of 3, 5, 7, 9, 11, and 13 mm, respectively, based on the Reference [59]. In addition, we also measured four median filters with quadratic kernels of 9, 25, 49, and 81 pixels.

From Table 3.2 showing the results by RMSE metric, it can be seen that for all statistics, the neural network based solutions achieved the best results. Note that under normal and 1/3 statistics, at this metric, the performance of the BM3D and Gaussian filters is comparable to the neural network, but with worse statistics, the performance of these solutions degrades to unusable levels. The performance of the filters calculated using the SSIM is shown in Table 3.3. This measure is not sensitive enough in case of good statistics, for which the score of the images is nearly equivalent. It can be seen, however, that with low statistics, BM3D performance is exceptionally bad when measured by SSIM, as well as when MAE is calculated, see Table 3.4. This is presumably because of this filter is designed to minimize the squared error. Neural networks have the best performance in case of all examined statistics.

A Figure 3.7 shows that the performance of the neural network is not only the best, but also has a low variance. Even for images with very low statistics, the scores of the different measures are close, unlike, for example, the median filter. Figure 3.8 with normal statistics and Figure 3.9 with 1/3 statistic show the performance evaluations of the best performing filters for different measures. It can be seen that SSIM and MAE move together, so the trend is different between the filters in terms of RMSE. In particular, for 1/3 statistics, the poor performance of BM3D using these metrics is evident.

3.4.2 Testing the robustness of the best performing neural network on specialized validation sets

In addition to the tests presented above, we were interested in the robustness of the neural network under different homogeneous, biased validation sets. We created groups based on commonly known criteria that can be reliably computed or accessed from information contained in DICOM files: women - men, elderly -

	Statistics									
	normal		1/3		1/9		1/16		1/32	
RMSE:	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
BM3D	**1.29**	0.36	**2.07**	0.33	4.50	0.36	7.34	0.46	13.76	0.72
Gaussian 11mm	1.99	0.93	2.21	0.91	**2.76**	0.83	3.27	0.82	4.21	0.81
Gaussian 13mm	2.34	1.21	2.48	1.19	2.85	1.12	**3.23**	1.10	**3.94**	1.05
Gaussian 3mm	2.81	0.33	4.85	0.57	8.38	0.97	11.16	1.31	15.80	1.85
Gaussian 5mm	1.69	0.31	2.66	0.37	4.44	0.53	5.87	0.70	8.26	0.97
Gaussian 7mm	1.56	0.47	2.15	0.45	3.33	0.48	4.30	0.57	5.97	0.73
Gaussian 9mm	1.71	0.68	2.07	0.65	2.87	0.60	3.58	0.63	4.82	0.69
L-NN	**1.15**	0.40	**1.38**	0.41	**1.80**	0.47	**2.09**	0.54	**2.54**	0.63
Median 3px	1.79	0.47	2.48	0.45	4.36	0.42	6.81	0.37	11.70	0.82
Median 5px	2.64	0.91	3.06	0.87	4.50	0.72	6.71	0.55	11.51	0.82
Median 7px	3.57	1.30	3.91	1.26	5.15	1.08	7.17	0.83	11.82	0.93
Median 9px	4.57	1.67	4.90	1.63	6.01	1.45	7.85	1.17	12.32	1.13
S-NN	1.21	0.35	1.56	0.38	2.09	0.48	2.45	0.56	3.00	0.67

Table 3.2: Performance of different filters calculated by RMSE. From the table, it can be seen that for all statistics, the neural network based solutions achieved the best results (smallest Mean and SD). Note that under normal and 1/3 statistics, at this metric, the performance of the BM3D and Gaussian filters is comparable to the neural network, but with worse statistics, the performance of these solutions degrades to unusable levels.

young, low-high body mass index (BMI) and created a mixed set as a reference benchmark dataset.

Based on the results shown in Table 3.5 and Table 3.6, it can be said that the performance of the L-NN, measured by both RMSE and MAE on different validation sets, is better than the results of S-NN for all sets and statistics. The trends in performance measured on the different sets as a function of the deterioration of the statistics are the same as those observed on the mixed set.

The performance of the L-NN filter can be seen in the box-plot type graph ([60], [61]) computing MAE on different validation sets in Figure 3.10. Median values are nearly the same for all data sets, the number of outliers is small, and the size of the interquartile intervals is comparable for all sets.

	Statistics									
	normal		1/3		1/9		1/16		1/32	
SSIM:	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
BM3D	9.88E-01	3.02E-03	9.34E-01	4.25E-03	6.75E-01	1.26E-02	4.85E-01	1.88E-02	3.33E-01	2.24E-02
Gaussian 11mm	9.92E-01	3.58E-03	**9.89E-01**	3.56E-03	9.82E-01	3.71E-03	9.73E-01	4.14E-03	9.57E-01	5.41E-03
Gaussian 13mm	9.90E-01	4.18E-03	9.88E-01	4.16E-03	**9.83E-01**	4.20E-03	**9.78E-01**	4.35E-03	**9.66E-01**	4.90E-03
Gaussian 3mm	9.70E-01	4.37E-03	9.27E-01	9.64E-03	8.54E-01	1.60E-02	8.09E-01	1.88E-02	7.57E-01	2.13E-02
Gaussian 5mm	9.90E-01	2.16E-03	9.75E-01	3.66E-03	9.39E-01	7.67E-03	9.08E-01	1.08E-02	8.60E-01	1.48E-02
Gaussian 7mm	9.93E-01	2.63E-03	9.85E-01	2.90E-03	9.65E-01	4.61E-03	9.46E-01	6.70E-03	9.11E-01	1.02E-02
Gaussian 9mm	**9.93E-01**	3.08E-03	9.88E-01	3.10E-03	9.77E-01	3.66E-03	9.64E-01	4.71E-03	9.40E-01	7.06E-03
L-NN	**9.96E-01**	3.02E-03	**9.94E-01**	3.34E-03	**9.92E-01**	3.59E-03	**9.90E-01**	3.82E-03	**9.86E-01**	4.19E-03
Median 5px	9.86E-01	4.97E-03	9.78E-01	4.96E-03	9.37E-01	9.44E-03	8.63E-01	2.56E-02	7.54E-01	3.02E-02
Median 7px	9.81E-01	6.93E-03	9.72E-01	6.98E-03	9.32E-01	9.99E-03	8.59E-01	2.57E-02	7.52E-01	2.97E-02
Median 9px	9.75E-01	8.77E-03	9.66E-01	8.73E-03	9.24E-01	1.04E-02	8.54E-01	2.53E-02	7.49E-01	2.87E-02
S-NN	9.95E-01	2.73E-03	9.93E-01	3.00E-03	9.89E-01	3.52E-03	9.86E-01	3.97E-03	9.80E-01	4.70E-03

Table 3.3: The performance of the filters is calculated using the SSIM. This measure is not sensitive enough for good statistics, for which the score of the images is nearly equivalent. However, it can be concluded that with low statistics, BM3D generates exceptionally bad images when measured by SSIM.

	Statistics									
	normal		1/3		1/9		1/16		1/32	
MAE:	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
BM3D	1.31	0.31	2.67	0.36	6.75	0.68	11.42	1.10	21.87	2.07
Gaussian 11mm	1.12	0.35	**1.32**	0.33	1.78	0.33	2.18	0.34	2.88	0.39
Gaussian 13mm	1.24	0.40	1.39	0.38	**1.73**	0.37	**2.04**	0.38	**2.60**	0.40
Gaussian 3mm	2.00	0.22	3.43	0.37	5.87	0.63	7.81	0.84	10.94	1.25
Gaussian 5mm	1.18	0.20	1.88	0.24	3.14	0.35	4.14	0.46	5.78	0.64
Gaussian 7mm	1.03	0.25	1.47	0.25	2.33	0.30	3.03	0.36	4.20	0.48
Gaussian 9mm	**1.03**	0.30	1.33	0.28	1.95	0.30	2.47	0.33	3.36	0.41
L-NN	**0.76**	0.31	**0.92**	0.32	**1.17**	0.33	**1.34**	0.35	**1.62**	0.37
Median 3px	1.16	0.28	1.74	0.26	3.22	0.27	5.26	0.36	9.40	0.81
Median 5px	1.42	0.40	1.85	0.38	3.14	0.36	5.09	0.40	9.23	0.79
Median 7px	1.77	0.52	2.17	0.52	3.39	0.47	5.27	0.45	9.38	0.81
Median 9px	2.19	0.65	2.59	0.65	3.76	0.60	5.58	0.53	9.63	0.85
S-NN	0.83	0.28	1.05	0.29	1.37	0.32	1.60	0.35	1.95	0.39

Table 3.4: Performance of different filters calculated as MAE. The performance of neural networks is the best for all statistics. For this metric, the performance of BM3D is much lower than for RMSE.

		Statistics									
		normal		1/3		1/9		1/16		1/32	
	RMSE:	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
L-NN	Age High	1.28	0.48	1.54	0.49	2.00	0.54	2.32	0.58	2.82	0.67
	Age Low	1.05	0.41	1.28	0.41	1.68	0.47	1.96	0.54	2.41	0.64
	BMI High	1.09	0.43	1.31	0.44	1.68	0.47	1.94	0.50	2.33	0.57
	BMI Low	1.28	0.48	1.54	0.48	2.02	0.54	2.35	0.61	2.89	0.72
	Female	1.20	0.43	1.43	0.44	1.83	0.48	2.12	0.53	2.55	0.61
	Male	1.20	0.45	1.46	0.46	1.91	0.54	2.23	0.61	2.71	0.73
	Mixed	1.15	0.40	1.38	0.41	1.80	0.47	2.09	0.54	2.54	0.63
S-NN	Age High	1.35	0.43	1.74	0.46	2.31	0.54	2.70	0.60	3.29	0.71
	Age Low	1.12	0.35	1.47	0.38	1.99	0.46	2.34	0.54	2.88	0.65
	BMI High	1.16	0.39	1.47	0.41	1.93	0.48	2.26	0.53	2.73	0.61
	BMI Low	1.34	0.42	1.76	0.45	2.36	0.55	2.78	0.62	3.42	0.74
	Female	1.27	0.38	1.62	0.41	2.14	0.48	2.50	0.55	3.02	0.64
	Male	1.27	0.41	1.65	0.44	2.21	0.56	2.59	0.63	3.17	0.77
	Mixed	1.21	0.35	1.56	0.38	2.09	0.48	2.45	0.56	3.00	0.67

Table 3.5: Performance of neural network-based filters computed by RMSE on different validation sets. The performance of the larger neural network is better than the smaller neural network for all sets and statistics. The trends in performance on different sets as a function of the degradation of the statistics are the same as those on the mixed set.

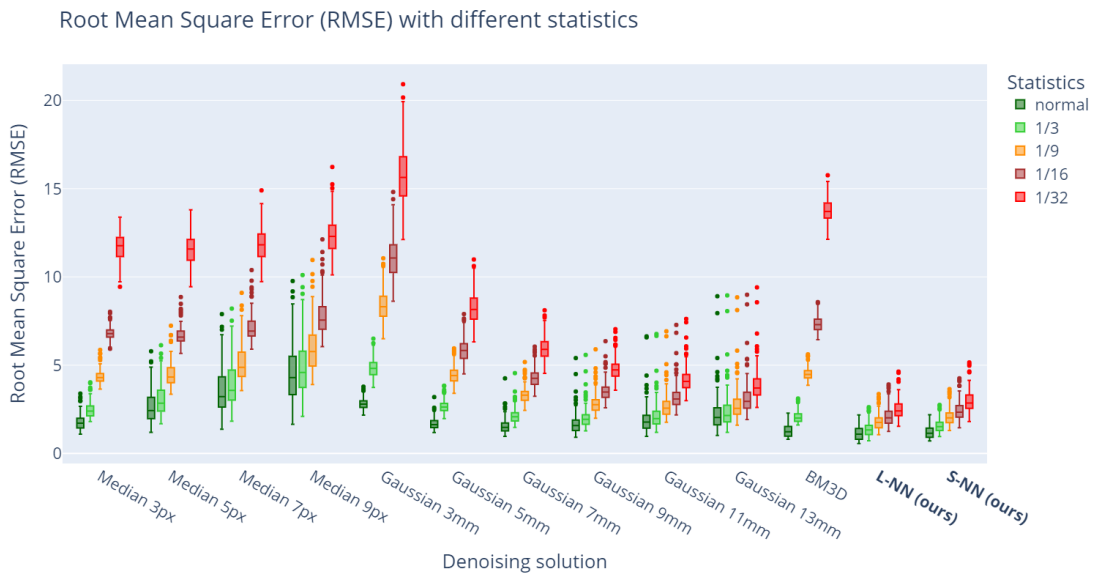
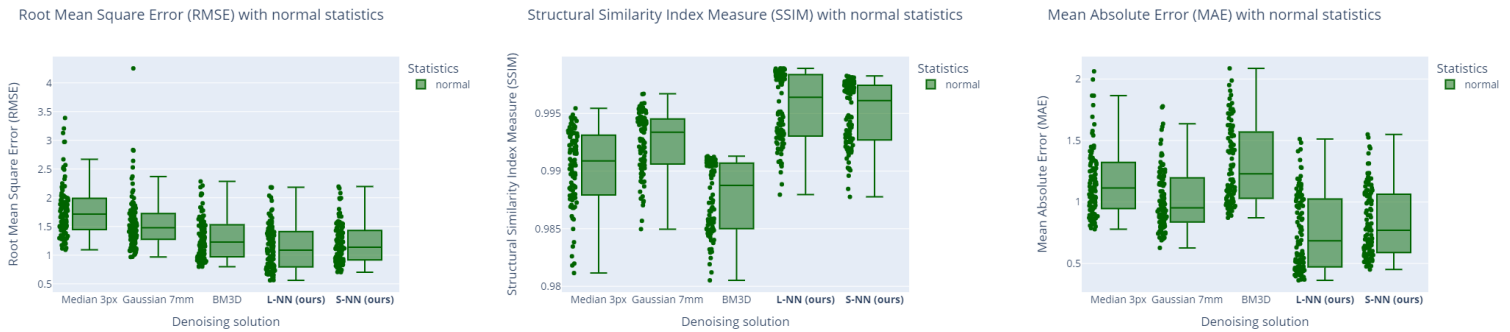


Figure 3.7: Performance of different filters for RMSE. The graph shows that the performance of the neural network, in addition to having the best values, also has a low standard deviation. Even for images with very low statistics, the scores of the different measures are close, unlike for example the median filter. For detailed description of the box plots see [60] and [61].

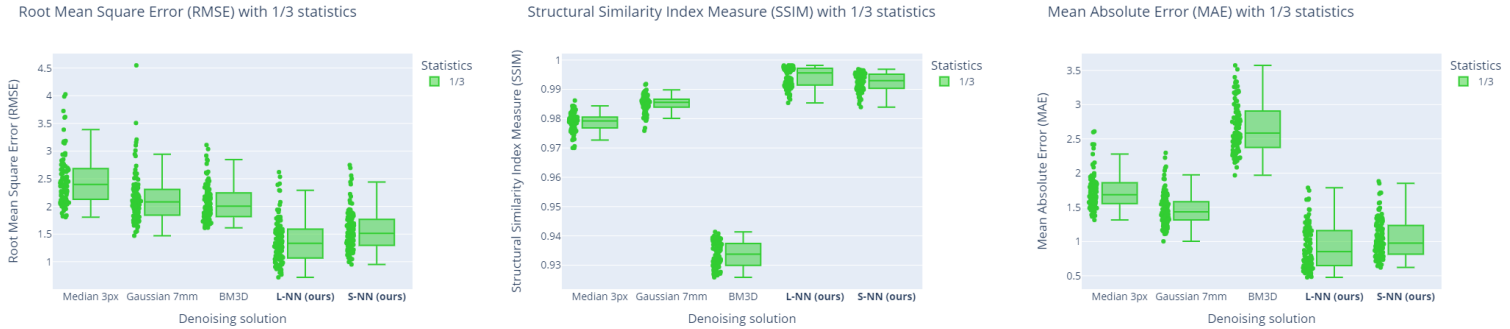
		Statistics									
		normal		1/3		1/9		1/16		1/32	
	MAE:	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
L-NN	Age High	0.86	0.38	1.04	0.39	1.31	0.40	1.50	0.42	1.80	0.45
	Age Low	0.68	0.30	0.83	0.30	1.06	0.31	1.23	0.33	1.51	0.36
	BMI High	0.76	0.35	0.92	0.35	1.14	0.36	1.30	0.37	1.55	0.39
	BMI Low	0.82	0.35	0.99	0.36	1.27	0.38	1.46	0.39	1.77	0.43
	Female	0.81	0.34	0.97	0.34	1.21	0.35	1.38	0.36	1.65	0.39
	Male	0.78	0.35	0.96	0.35	1.23	0.37	1.41	0.39	1.70	0.42
	Mixed	0.76	0.31	0.92	0.32	1.17	0.33	1.34	0.35	1.62	0.37
S-NN	Age High	0.93	0.35	1.17	0.36	1.52	0.40	1.77	0.42	2.15	0.46
	Age Low	0.75	0.26	0.96	0.27	1.27	0.29	1.49	0.32	1.83	0.36
	BMI High	0.83	0.32	1.03	0.33	1.32	0.35	1.53	0.38	1.85	0.41
	BMI Low	0.89	0.32	1.13	0.33	1.49	0.37	1.75	0.39	2.14	0.44
	Female	0.88	0.30	1.10	0.31	1.41	0.34	1.64	0.36	1.99	0.40
	Male	0.86	0.31	1.09	0.33	1.43	0.37	1.67	0.39	2.04	0.45
	Mixed	0.83	0.28	1.05	0.29	1.37	0.32	1.60	0.35	1.95	0.39

Table 3.6: Performance of neural network-based filters computing MAE on different validation sets. The performance of the larger neural network is better than the smaller neural network for all sets and statistics. The trends in performance on different sets as a function of the degradation of the statistics are the same as those on the mixed set.



(a) RMSE calculated on **normal** statistics (b) SSIM calculated on **normal** statistics (c) MAE calculated on **normal** statistics

Figure 3.8: This graph shows the performance of the best performing filters at different metrics. It can be seen that SSIM and MAE move together, so the trend is different between filters in terms of RMSE.



(a) RMSE calculated on **1/3** statistics (b) SSIM calculated on **1/3** statistics (c) MAE calculated on **1/3** statistics

Figure 3.9: This graph shows the performance of the best performing filters at different metrics. It can be seen that SSIM and MAE move together, so the trend in RMSE is different between the filters and the difference is much larger than for normal statistics.

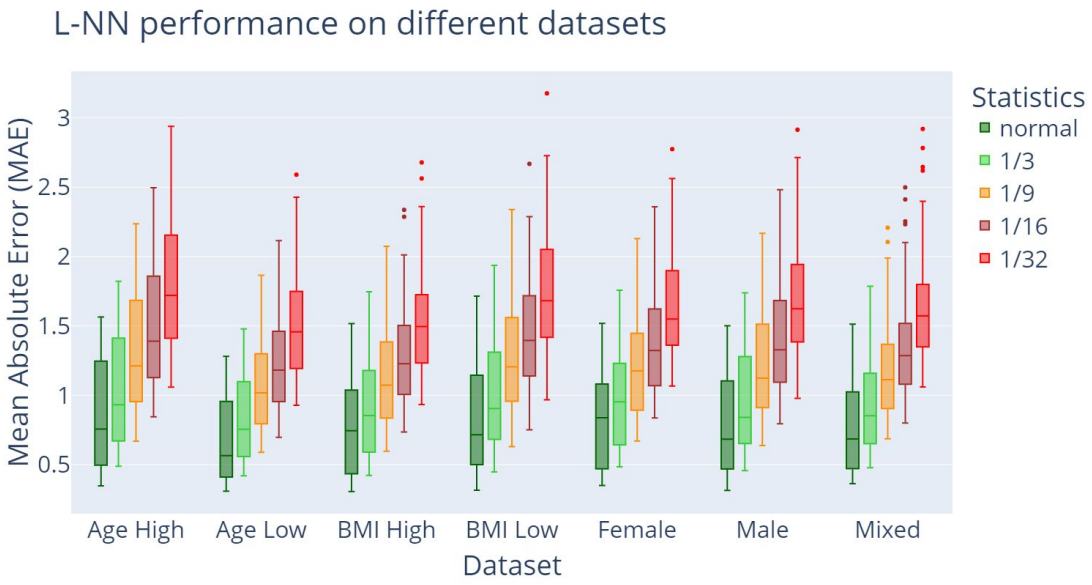


Figure 3.10: The performance of the L-NN filter computing MAE on different validation sets. The box-plot type graph shows the measured performance for each data set. It can be seen that the median values are nearly the same for all data sets, with a small number of outliers. For detailed description of the box plots see [60] and [61].

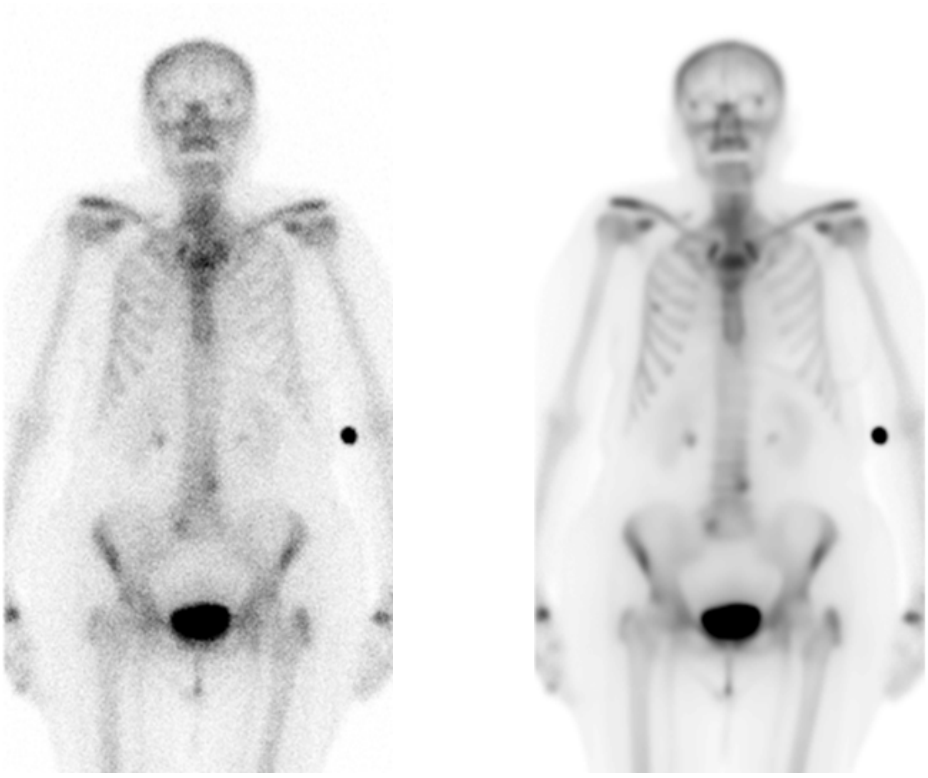
3.4.3 Preliminary clinical evaluation

After the robustness test, a study of clinical pre-testing had been accomplished involving physicians (ScanoMed Ltd., Debrecen, Hungary). The aim of this study was to allow doctors who have worked with many similar images to point out possible defects, artificial products and to give their opinion on the usability of the device. The images of 412 routine bone scintigraphy whole-body examinations at ScanoMed were denoised using the AI-based application presented here. Patients routinely received 550-600 MBq of 99mTc-MDP intravenously, and whole-body images were acquired after 2 hour of accumulation time. Once the planar image was acquired, the filtered image was obtained within 1-2 minutes and helped physicians to decide on additional investigations such that if any image showed a lesion suspicious for metastasis, SPECT/CT was indicated. As we reported in the Reference [41], the doctors looked at the original unmodified image with normal statistics and the noise-filtered version of the image in parallel, and evaluated the images in this way. The physicians found that the neural network based filter did not delete or generate new lesions, and they don't identified artifacts on the pictures. They concluded, that it was easier to localize the abnormalities (count ribs, vertebrae), decide whether additional examinations (SPECT/CT) was needed, and all this accelerated the diagnosis itself.

This experiment suggests that the use of a noise filter is useful for images with normal statistics, but further studies are needed to see how much it is possible to reduce the measurement time or the activity administered preserving the original, reliable diagnostic capability.

In the provided figure (Figure 3.11), a visual representation of bone scintigraphy images before and after the application of a denoising algorithm is presented. The figure consists of two side-by-side images, wherein the first image showcases the initial input bone scintigraphy data, capturing the raw information obtained during the imaging process. This raw image, serving as the baseline, reflects the original quality and clarity of the acquired data. The second image displayed adjacent to the raw input is the denoised output obtained after the application of the developed denoising algorithm. This denoised version highlights the substantial improvement achieved in image quality, showcasing enhanced clarity, reduced noise

artifacts, and improved delineation of anatomical structures. The comparison between the raw input and denoised output images serves to underscore the efficacy and transformative impact of the denoising algorithm in enhancing the interpretability and diagnostic utility of bone scintigraphy images.

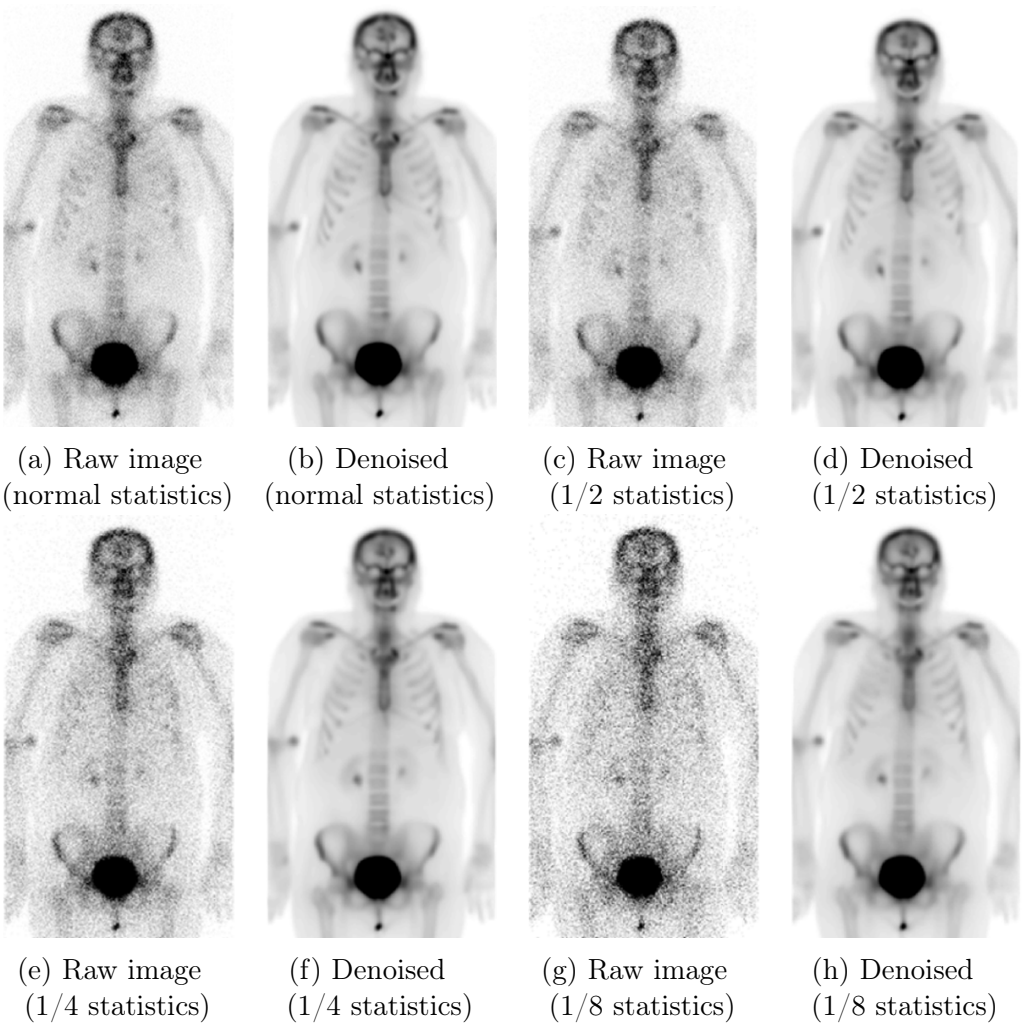


(a) Input (b) Output

Figure 3.11: This figure presents a comparative view of two images related to bone scintigraphy. The first image (a) depicts the initial input scan, capturing the raw data acquired during the imaging process. The second image (b) exhibits the output after the application of the developed denoising algorithm (L-NN).

In Figure 3.12, a visual representation of the impact of varying statistical reductions on bone scintigraphy images and their subsequent denoising outcomes is presented. The series of eight images systematically demonstrates the effect of decreasing statistical parameters on the raw images, simulating scenarios with reduced data quality. The first four images depict pairs of raw bone scintigraphy images: the first with normal statistical parameters and the second with deliberate reductions to 1/2 of the statistics. Corresponding denoised versions of

these images demonstrate the efficacy of the denoising algorithm under differing data quality conditions. Additionally, the series extends to include images with statistical reductions to 1/4 and 1/8 of the original parameters, followed by their respective denoised counterparts. This comprehensive visual analysis serves to illustrate the considerable impact of data quality variations on the denoising process, emphasizing the algorithm's ability to enhance image quality even under significantly reduced statistical conditions.



(a) Raw image (normal statistics) (b) Denoised (normal statistics) (c) Raw image (1/2 statistics) (d) Denoised (1/2 statistics)

(e) Raw image (1/4 statistics) (f) Denoised (1/4 statistics) (g) Raw image (1/8 statistics) (h) Denoised (1/8 statistics)

Figure 3.12: This figure showcases a series of eight images depicting the original and denoised versions (created by L-NN) of bone scintigraphy under different statistical reductions, highlighting the impact of data quality on the denoising process.

Therefore we have been working on the complex clinical evaluation of the given denoising algorithm integrated with lesion detection and classification software components in order to optimize the performance regarding ROC (receiver operation curve) analysis. Our future aim is to ensure clinical diagnostic value regarding sensitivity and specificity even at significantly lower administered activities or measurement time using the presented denoising solution.

3.5 Conclusion

We have demonstrated that it is possible to train a neural network that performs well under a wide range of noise levels and outperforms previous non-neural network based tools such as Gaussian filter, median filter and BM3D. Noise-filtered images may allow to reduce the amount of injected activity and the measurement time, and may also improve the accuracy, speed and reliability of diagnosis, but this must be supported by clinical trials. Such a noise-filtering solution can also be used to improve the image quality of fast, localisation preview scans. The evaluation method presented here can be applied and generalised in all cases where noise-free measurements are not available.

Chapter 4

Proposed loss function for neural network based segmentation: wave loss

4.1 Comparison of Shapes and the Binary Wave Metric

In every application currently non-topographic metrics are applied to calculate pixel-wise differences between images, which completely neglects topographic information. In this section, we focus on binary (black and white images) to illustrate the flaw of pixel-based metrics and reveal how the wave metric can enhance these similarity functions.

In a particular problem, a metric has to be chosen depending on the nature of the problem. All metrics are problem-dependent and since a metric condenses high-dimensional similarities into a scalar value, no metric can be general and perform well for every practical problem. This motivates us to use topographic metrics for topographic problems, like segmentation.

The most commonly applied metrics for binary objects are Hamming [62] and Hausdorff [63] distances.

Hamming distance computes the number of differing pixels between two images:

$$D_{Hm} = \sum (A \bigcup B) \setminus (A \bigcap B) \tag{4.1}$$

where A and B are the input images both containing only values of zeros and ones.

This metric is fast and easy to calculate and although it is commonly applied to compare the shapes of various objects, in many tasks it performs poorly because of a complete lack of topological information. This metric is also commonly referred as an area-based metric since only the area of the differing regions determines the metric and it takes into account the number of different pixels regardless of their neighbors or their relative positions. Almost all popularly used metrics such as cross entropy, Dice [64], Lovász [65] or Tversky [66] losses are area-based metrics, where the area of the different regions matters; their topologies are not considered.

In the case of grayscale images, an extension of this metric can be applied as the pixel-wise difference between the two images; these metrics are usually referred to as ℓ_1 and ℓ_2 distances and can be defined the following way:

$$\ell_1(a,b) = \sum_i |a_i - b_i| \tag{4.2}$$

and

$$\ell_2(a,b) := \sqrt{\sum_i (a_i - b_i)^2} \tag{4.3}$$

Figure 4.1 demonstrates the contradiction between the Hamming distance and subjective human judgments. However, every human observer would judge the middle-right pair to be more similar; the Hamming distances between the middle and left and the distance between the middle and right images are exactly the same. Our perception is based not only on the area of the differing parts, but also on shape-related information. This information cannot be ignored if we want to create a trustworthy metric. One can see that the Hamming distance (and any other area-based metric) can provide misleading decisions. Thus, shape-related descriptors are also required.

(a) (b) (c)

Figure 4.1: Two images and a reference image with the same Hamming distances but different topology; (**a**) compared image 1; (**b**) reference image; (**c**) compared image 2.

The other often used metric is the Hausdorff distance [63], which is determined solely by the distance of the furthest different pixel between the objects:

$$D_{Hs} = max(h(A, B), h(B, A)) \tag{4.4}$$

where $h(F, G) = \max_{f \in F} \min_{g \in G} d(f, g)$, d is a distance measure (e.g., ℓ_1 or ℓ_2), F and G are the images to be compared and f and g represent their pixels. This distance represents topology better since Hamming distance is determined only by the area of the different pixels; meanwhile, here it ensures that all differences are in a D_{HS} radius. Since this metric can reflect topology better, it is employed in certain applications such as geology [67] and quantum physics [68]. Unfortunately, this metric's sensitivity to noise prevents its utilization in practical applications. If the largest distances between two different pixels are the same on two image pairs, the metric will return the same result and hides all other information about shapes as well.

The illustration of the Hausdorff distance and its poor applicability can be seen in Figure 4.2. This image illustrates the contradiction between the Hausdorff distance and subjective judgments. Hausdorff distances between the image in the middle and the one to the left and between the image in the middle and the one to the right are exactly the same. A human observer would most probably select the middle-left pair as more similar images. Natural vision and perception is not only based on the topology of the differing parts, but also on area-based information. This information also has to be considered during computation to produce a reliable and useful metric.

(a) (b) (c)

Figure 4.2: Three images with the same Hausdorff distances but different topology; (**a**) compared image 1; (**b**) reference image; (**c**) compared image 2.

Binary Wave Metric

Both the Hamming and Hausdorff metrics reveal important properties about similarity, but to create a multipurpose, efficient metric, their advantageous properties should be combined, eliminating their flaws. The application of different metrics in a parallel manner might be beneficial, but in the case of simultaneously computed metrics one will increase processing time and algorithmic complexity and we may not be able to solve the problem since the weighting of these metrics during combination is always problematic. One has eventually to combine all the metrics into a single function that results in a scalar to ease classification and comparison.

The idea of the binary wave metric was first introduced by Istvan Szatmari in 1999 [69]. His work covers the metric calculation for convex, binary objects only and in this work we will extend it to non-convex, grayscale images, which makes it applicable as a loss function in image segmentation algorithms. This metric can be defined as the volume of an ascending wave starting from the intersection of the objects and filling out the area defined by the union of the two binary objects.

On a suitable hardware architecture, the non-linear wave metric can measure both the shape and the area difference between two objects in a single operation (e.g., on a multi-layer cellular neural network) [70].

Based on this, the equation of the metric calculation for convex two-dimensional binary objects can be given as the following:

$$W_M(A,B) = \int_{x,y \in S_{Hm}(A,B)} D_{Hs}(x,y) \tag{4.5}$$

which is the point-wise integration of local Hausdorff distances over every point in the disjunctive union of the two objects. S_{Hm} defines the pixels where $D_{Hm} > 0$. In the case of non-convex objects, the non-linear wave metric is the integration of the local Hausdorff distances along the shortest path in the union of the two sets.

It can be easily seen that the wave metric contains and compresses both previously introduced metrics. The maximal height of the ascending wave (the propagation time of the wave) is proportional to the value of the Hausdorff metric (D_{Hs}) for connected objects and the area of the wave propagation is proportional to the Hamming distance (D_{Hm}).

The slope, the increase of the wave for each step, determines the connection between the topological (Hausdorff) and the area-based (Hamming) information. For example, if this increase is set to zero and propagation starts with a constant non-zero magnitude, the wave metric will yield the Hamming distance multiplied by the initial constant. In the case of a larger slope, the metric will shift more towards the Hausdorff metric and the distances between the further and further differing points will determine the result more and more.

It is easy to see that this similarity function fulfills almost all the required properties of a metric. It can be defined as a function on a given set $d : X \times X \mapsto \mathbb{R}$ and it fulfills the following properties ($\forall a, b, c \in X$): non-negativiy or separation axiom: $d(a, b) \geq 0$; identity of indiscernibles, or coincidence axiom: $d(a, b) = 0 \Leftrightarrow a = b$; and symmetry: $d(a, b) = d(b, a)$.

Unfortunately, the triangle inequality or subadditivity axiom

$$d(a, c) \leq d(a, b) + d(b, c)$$

does not hold this way since three objects, from which two are completely disjoint (a and c) and the third of which has a common part with both (b), would result in a zero value for $d(a, c)$ and a non-zero value for both $d(a, b)$ and $d(b, c)$. To get around this problem, we applied an extra penalty for points which cannot be reached in the union during wave propagation. This penalty has to be larger than the maximum penalty in the reachable region. With this addition, the wave metric fulfills all the axioms and forms a proper metric for non-connected objects as well.

The illustration of the wave propagation and the metric can be seen in Figure 4.3. In this image, the first row depicts two possible binary input images (first and second images from the left) and their intersection and union (third and fourth images from the left). The last two rows depict four 3D versions of the wave metric in an increasing manner until reaching the union at iterations 100, 150, 300 and the last iteration, including not reached regions. During propagation, further and further pixels will be incorporated in the loss function with higher and higher values. At the last step, a high penalty will be assigned to all pixel which were not reached during propagation.

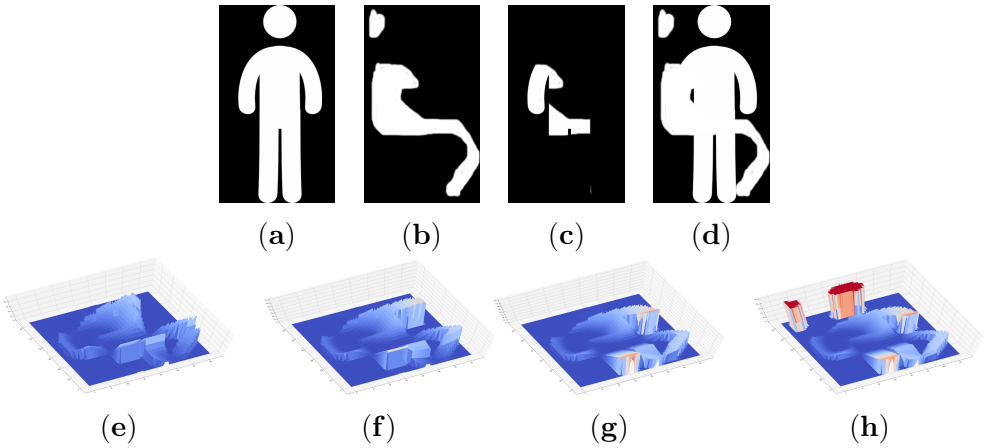


Figure 4.3: Illustration of the wave propagation; (**a**) input A; (**b**) input B; (**c**) intersection; (**d**) union; (**e**) wave 100 iterations; (**f**) wave 150 iterations; (**g**) wave 300 iterations; (**h**) wave unreached regions.

This metric can be used to compare images and accurate usable calculations, but unfortunately cannot be used during network training, as was shown in [71], since it can only be calculated between two binary images.

4.2 Wave loss: Extension of the Wave Metric to Three Dimensions

In the previous section, the binary wave metric was described; as was demonstrated, it creates a connection between topological and area-based metrics. To extend it to grayscale images and two-dimensional probability distributions, we have to consider intensity-based differences as well. The metric should depend on three not-independent measures: the area of the differences, the topology of the differences and the intensities and values of the differences.

In this case the output and ground-truth images can be imagined as two two-dimensional surfaces in three dimensions. From this, we can calculate the intersection and the union (which will also be two-dimensional surfaces); then, the metric can be imagined as a three-dimensional wave propagating and filling out the space between these two surfaces. A weight will be associated to every new voxel at each time step of the propagation and this four-dimensional volume (the weighted sum of the three-dimensional changes) will be called wave loss.

Our goal was to differentiate between value and topology-based differences and because of this the propagation speed of the wave is different in z (intensity) and x, y (topological directions) (the wave could propagate differently along the x and y dimensions as well, but in image-processing applications these dimensions are usually handled in the same manner).

Compared to the binary wave metric, where only topological distances were covered, an upper bound for the number of required steps until convergence can easily be identified. An upper bound for spatial propagation can also be found (identifying the object containing the longest possible path with the given image size), but this bound is fairly high compared to the number of steps required to cover differences in intensity. In practice, this means that typically a small number of iterations (10–20) is enough to calculate the metric.

Algorithm 1 calculates wave loss for two grayscale images. The input values are $Img1$ and $Img2$ and the output of the algorithm is a scalar variable $WaveLoss$. The parameters of the algorithm are the following:

- $ValInc$ will determine how fast the wave propagates along the intensity differences; every pixel's intensity will be increased by this amount in every iteration. This parameter will also determine the maximum number of required iterations and by this it will also determine the largest distance from the intersection where topological differences are considered. Having a larger distance than the maximal receptive field of a neuron in the network is illogical because this way the error could be derived back to a neuron which had no vote in the classification of that input pixel. In our experiments, this value was between 0.05 and 0.1, meaning that the wave from a selected point could propagate for 20 and 10 pixels.

- $SpaInc$ will determine the spatial propagation speed of the wave. Spatial propagation is implemented by a max pooling operation with window size $SpaInc$ and a stride of one. In our simulations, this value was always set to 3.

- $ValW$ is a vector of penalties for the intensity differences. If this value is constant, the weight differences will be linearly proportional to the penalties

in the loss. If this is increasing, it means larger differences (where more iterations are required to reach the desired value) will have larger and larger penalties. In our simulations, we used constant values in $ValW$.

- $SpaW$ is a vector containing the penalties for topographical differences. $SpaW[0]$ will weight those points which can be reached in one spatial propagation and which are in the direct neighborhood of the intersection. $SpaW[k]$ will have a penalty for those values which will be reached at the k-th iteration. In our simulations, we applied linearly increasing values which were all lower than the values of $ValW$. In most networks, we want to have good results on average, but minor mistakes about the shape of the object can be tolerated. Applying lower values than the intensity weights ($ValW$) means that the importance of the shape of the segmented object will become less important. Monotonically increasing $SpaW$ means that the further we are from the object, the higher the cost a misclassification will result. Applying higher weights than $ValW$, which are monotonically decreasing, would mean that the boundaries are really important and classifying a pixel around a boundary is a larger problem than misclassifying a pixel somewhere far from the object.

Since both the values in $Img1$ and $Img2$ are bounded, the algorithm will always converge if a larger than zero $ValInc$ parameter is applied. The difference between the intersection and the union will decrease at least by $ValInc$ amount in every iteration. Since both the intersection and the union are probability distributions in the case of neural network training (just like $Img1$ and $Img2$) we can consider their values to be between zero and one and this way the algorithm will always converge in $1/ValInc$ iterations. One can easily see that the wave metric is an extension of the normal ℓ_1 metric; if there is no spatial propagation ($SpaInc = 0$) and ValW values are all the same, we will obtain L1 loss as a result.Similarly, if the values in ValW are increasing exponentially with no spatial propagation, this metric will calculate the traditional cross entropy between the two images, representing two-dimensional distributions.

The derivative of Algorithm 1 is also required for network training. Luckily, our method consists of simple operations such as addition, multiplication and

Algorithm 1: Calculation of wave loss.

Data: Img1, Img2
Parameters : ValInc, SpaInc, SpaW, ValW
Result: WaveLoss

```
Union ← max(Img1,Img2);
CurrentWave ← min(Img1,Img2);
NewWave ← min(Img1,Img2);
WaveLoss = 0;
i ← 0;
num_iter ← int(1/ValInc);
while i ≤ num_iter do
    /* Loss for intensity differences                    */
    NewWave += ValInc;
    NewWave = min(NewWave,Union);
    ValueChange = sum(NewWave-CurrentWave);
    WaveLoss += ValW[i]*ValueChange;
    CurrentWave = NewWave;
    /* Loss for spatial differences                      */
     NewWave = maxpool(CurrentWave,[SpaInc,SpaInc], [1,1]);
    NewWave = min(NewWave,Union);
    SpatialChange = sum(NewWave - CurrentWave);
    WaveLoss += SpaW[i] * SpatialChange;
    CurrentWave = NewWave;
    i += ValInc;
end
```

maximum/minimum selection. The derivative of all of these individual operations (each line in the algorithm) can be calculated in a straightforward manner and the derivative of the algorithm can be determined by the chain rule. Luckily, in modern machine-learning frameworks such as Pytroch [72] or Tensorflow [51] where automatic differentiation is applied, these derivatives are calculated automatically.

During the calculations, we first increase $CurrentWave$ according to the intensities. This is a global change and it happens everywhere in the image where the values have not reached the union and after this step we apply spatial propagation. One could change this order, but we considered intensity-based differences more important. One could also execute both propagations separately and sum their penalties, but we did not observe any measurable effect applying this modification. In this setup, compared to the binary implementation all points are reached during propagation since the intensity differences are limited; therefore, there is no need for an extra penalty for unreached regions.

Since it is difficult to plot wave loss using two-dimensional images, we opted to display it using two one-dimensional grayscale 'images'. This can be seen in Figure 4.4. As can be seen, the wave fills out the region between the intersection and the union of the two surfaces. At each propagation, the newly reached pixels will be weighted and added to the loss function. This way, this metric incorporates intensity-, area- and shape-related information simultaneously.

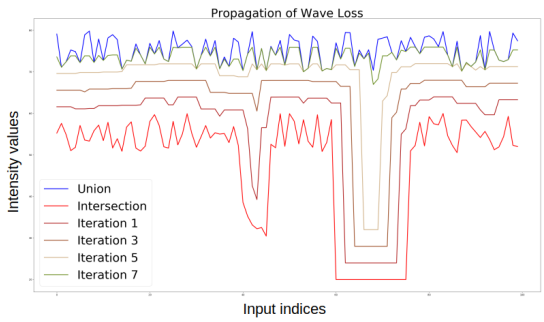


Figure 4.4: The propagation of the wave during the calculation of wave loss.

From an implementation point of view, value increase is just an addition and spatial propagation is a grayscale dilation, which is essentially a max pooling

operation which can be found in every modern machine-learning environment. For a training step, the number of additional pooling operations is fairly small compared to the pooling operations already contained by a typical convolutional neural network. Therefore, the calculation of the wave loss will not increase training time significantly and has no effect on inference time.

4.3 Materials and Methods

We used the Pytorch [72] open source machine-learning framework for the implementation of our algorithm and for training various neural network architectures on multiple datasets. For evaluation, we investigated three publicly available and commonly cited datasets: CLEVR [73], Cityscapes [74] and MS-COCO [75]. In our instance segmentation experiments on MS-COCO, we used the Detectron 2 environment [76]. For the sake of reproducibility and a detailed description of parameter setting, our code for network training and evaluation on both datasets along with the data generation script for the CLEVR dataset can be found at `https://github.com/horan85/waveloss` (accessed on the 28th of May 2022).

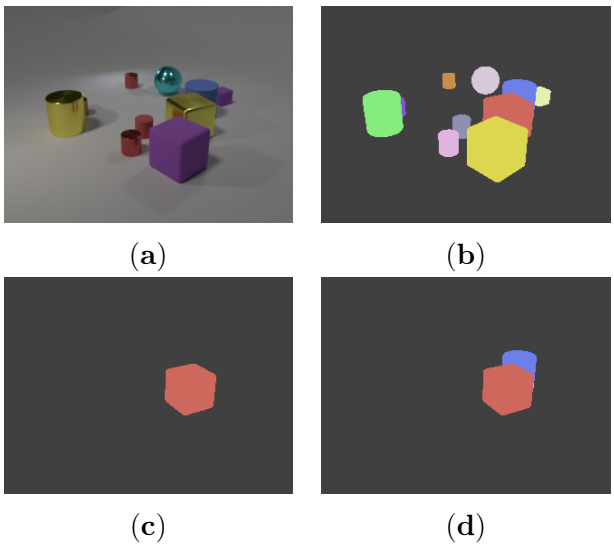
4.3.1 Simple Dataset for Segmentation

Since we were not able to find a simple segmentation dataset (like MNIST [77] or CIFAR for classification), we created a simple dataset based on CLEVR [73].

The dataset contains 25,200 three-channel RGB images (of size 320×240) of simple objects along with their instance masks, amodal masks and pairwise occlusions and three-dimensional coordinates for each object. These images were created in a simulated environment and contain cylinders, spheres and cubes in various positions. Since the dataset is simulated, the exact location and the pixel-based segmentation maps of all the images are known. This results in a simple dataset for various tasks, like three-dimensional reconstruction, instance segmentation and amodal segmentation.

The dataset contains objects of simple shapes, but also contains shadows, reflections and different illuminations, which make it relevant for the evaluation of segmentation algorithms.

An example image of the dataset along with a few generated masks can be seen in Figure 4.5. An input image (top left), the instance segmentation mask (top right), an example amodal mask which was generated for each object individually (bottom left) and the pairwise occlusion mask (bottom right) are displayed in this figure. The pairwise occlusion images with the amodal mask can be used to determine front–back relation between objects. Apart from the mask, the exact object coordinates and sizes are also stored in JSON format.



(**a**) (**b**)

(**c**) (**d**)

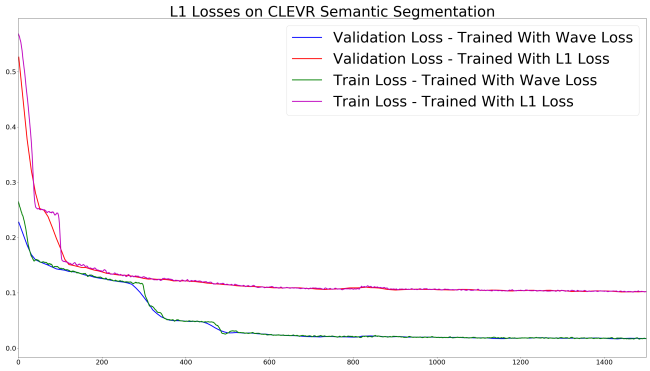
Figure 4.5: Example images from the CLEVR dataset; (**a**) input Image; (**b**) segmentation mask; (**c**) instance mask; (**d**) occlusion mask.

A simple simulated dataset: CLEVR

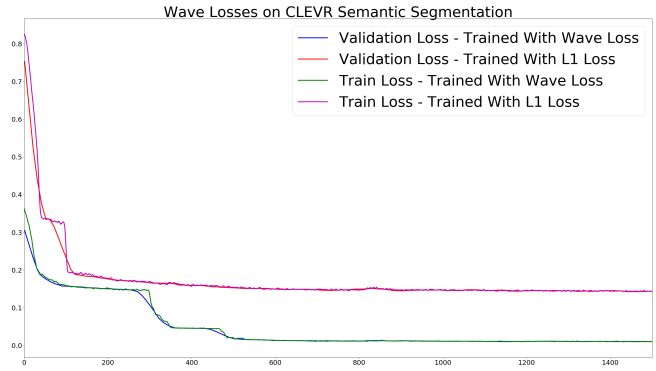
We selected the U-net architecture to compare wave loss and L1 loss in our simple CLEVR-inspired dataset. We used a U-NET-like structure containing 8, 16, 32, 64 convolution blocks (each 3×3). Downscaling was carried out by strided convolutions, while upscaling was implemented by transposed convolutions.

We trained the network 20 times independently on our dataset for semantic segmentation, using 23,400 images for training and 1800 images for validation (all the validation scenes were generated independently from the training scenes). In one setup, we trained the network to minimize the L1 loss on the training set, in the other setup wave loss was defined as the error function. In both cases, we measured both L1 and wave loss during training and validation. The losses can

be seen in Figure 4.6. Some qualitative examples from the validation set during different train iterations can be seen in Figure 4.7. The images were taken at 200, 400, 600, 1000 iterations from the training set. From left to right, the columns are the following: input image, network output trained with L1 loss, input image, network output trained with wave loss. One could also observe a formation of a spotted, grid-like structure at the first iterations of training using the wave loss since regions around high intensity pixels cause lower loss values.



(**a**)



(**b**)

Figure 4.6: Losses on the CLEVR dataset averaged out on 20 independent runs; (**a**) L1 loss; (**b**) wave loss.

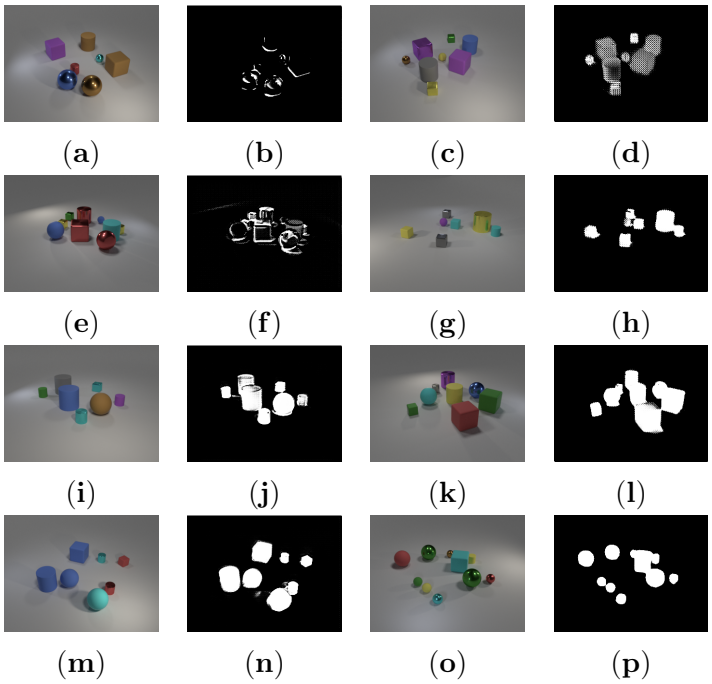


Figure 4.7: Example images at different iterations from the CLEVR dataset; (**a**) input; (**b**) seg. 200; (**c**) input; (**d**) seg. 200; (**e**) input; (**f**) seg. 400; (**g**) input; (**h**) seg. 400; (**i**) input; (**j**) seg. 600; (**k**) input; (**l**) seg. 600; (**m**) input; (**n**) seg. 1000; (**o**) input; (**p**) seg. 1000.

As one can see from these measurements, wave loss results in faster convergence and better accuracy in both the training and the validation sets.

These simple examples show the applicability of wave loss in segmentation tasks, but to demonstrate the advantage of this topological loss function, we have to investigate it in more complex and practical tasks.

4.3.2 Competitor loss functions

We have compared our method to a recent reformulation of Dice loss [64], which calculates the ratio between the intersection and the union between the two binary objects. Unfortunately, Dice loss is a metric applied over binary images and they are based solely on area-based differences. Until the number of pixels in the intersection and union remain the same, the regions can change arbitrarily. Namely, the different pixels can move anywhere in the image space; only the

number of different pixels matters. Another recent improvement over the area-based metric is the active boundary loss [78] where, as an additional loss value, the boundary pixels are calculated with a larger weight, this way representing the shape of the object in the loss function. This is more similar to our approach, but it considers only the boundary pixels and no other pixels in the differing area. The third selected loss is the shape aware loss function [79] which considers all pixels in a differing region, but with a precomputed weight which is the pixels' Euclidean distance from the intersection. Unfortunately, this distance is not the same as the shortest path of differing pixels since in the case of non-convex regions this distance can be significantly larger.

4.4 Results

4.4.1 Semantic segmentation on Cityscapes

We investigated the Cityscapes dataset [74] with the following architectures: SegNet [29], HRNET [80], DeepLab [81] and DeepLabv3 [82].

We trained these networks with three different loss functions (ℓ_1, cross entropy and wave loss) and the accuracy results can be found in Table 4.1.

During training, we initialized the weights randomly and executed five independent trainings with every configuration and trained them for 400,000 iterations.

The parameters of our loss function were the following: SpaInc was set to three; this means propagation happened using 3×3 kernels. Topology weights (SpaW) exponentially increased from 0.01 to 1 and intensity weights (ValW) were all set to a constant value of one. ValInc was set to 0.05; this means that the largest value gap of one will be filled in twenty iterations. Using this value, neighborhoods of maximum twenty pixels are affected by wave propagation. Since the input resolution of the networks was 513×513, we considered these 20×20 neighbourhoods sufficiently large.

As can be seen from the results, the application of wave loss increased the network performance compared to traditionally used cross entropy loss with an approximated 3% in the case of all network architectures and provided better segmentation accuracy than any of the investigated loss functions in all cases.

Table 4.1: This table contains the average accuracy results of five independent runs on the Cityscapes dataset using four different network architectures (rows) and six different loss functions for semantic segmentation.

Model	L1 Loss	CrossEnt	Dice	Boundary	ShapeAware	Wave
SegNet	54.2%	57.0%	57.3%	57.7	58.6%	**59.5**%
DeepLab	59.7%	63.1%	64.1%	64.3%	65.4%	**66.7**%
DeepLabv3	77.6%	81.3%	81.4%	81.5%	81.7%	**82.2**%
HRNET	77.4%	81.6%	81.8%	81.8%	82.1%	**83.4**%

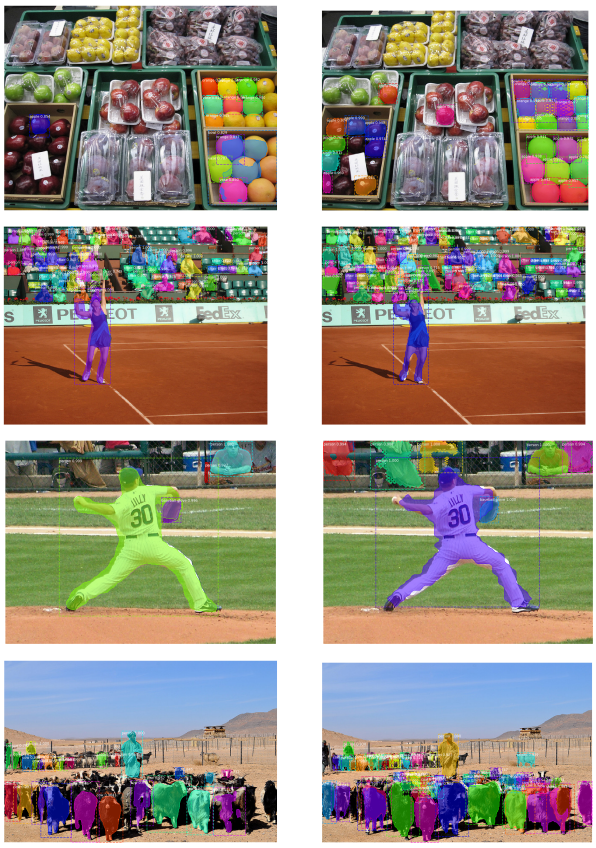
4.4.2 Instance segmentation on MS-COCO

We also investigated the problem of instance segmentation on COCO 2017 [75]. We investigated MASK R-CNN with different backbone architectures using the Detectron 2 framework where we kept the architecture and all the other parameters unchanged in the configuration files used for instance segmentation on this dataset. These configurations contained data augmentation in the input samples containing random flip, random crop, brightness change and random additive noise. The original training script used cross entropy and we added our implementation of the wave loss to the framework and compared its performance.

The parameters of our loss function were the same as in the case of the Cityscapes dataset. We would like to emphasize that the images used for segmentation differ significantly in size from the images used in Cityscapes since in the case of Mask R-CNN the segmentation head is executed on the 28×28 outputs of the RoiAlign layer. Even though the object sizes may differ, the same parametrization worked well for this architecture and dataset as well, which demonstrate that our loss function is not heavily dependent on the exact parameter values.

We measured mean average precision values using the evaluation script of COCO. We have to note that IOU is more related to wave loss than to L1 metric or cross entropy since wave loss uses the intersection and union to determine wave propagation, but we think this does not bring an unfair bias to the evaluation. The results can be seen in Table 4.2. As can be seen from the results, the application of the wave loss increased the precision of the network with an overall 3% on our validation set and the network performs especially better in the case of small objects, where an improvement of 5% was achieved compared to our

reference network trained by cross entropy loss. We also have to emphasize that segmentation improved in the case of every architecture and for all object sizes. Qualitative results about the generated masks and bounding boxes can be seen in Figure 4.8. The segmentation masks in the first column were generated by a network trained with cross entropy loss; the masks in the second column are results of a network trained with wave loss. As can be seen, wave loss does not cover boundaries as sharply, but altogether gives a better coverage of the objects. (Although we have to note that this judgment might be subjective and could also depend on the exact parametrization of wave loss.)



(**a**) (**b**)

Figure 4.8: Example results on the COCO dataset segmented with Mask-RCNN; (**a**) cross entropy loss; (**b**) wave loss.

Table 4.2: Average precision results on COCO 2017 validation set using the same network architectures with three different loss functions in different columns (ℓ_1, cross entropy, Dice loss, active boundary loss, shape aware loss and wave loss). Two different architectures (ResNet-50 and ResNet-101) can be found in the rows, with feature pyramid networks (FPNs) or when the activation of the fourth convolution layer (C4) was used for region proposals. The results display the mean average precision for all objects, except the last three rows, where the accuracy results for the best performing network are detailed for small-, medium- and large-sized objects as well.

Model	**L1**	**CrossEnt**	**Dice**	**Boundary**	**Shape**	**Wave**
R50-C4 mAP all	28.75%	32.2%	32.83%	32.9%	34.721%	**35.93**%
R50-FPN mAP all	29.43%	35.2%	36.14%	36.12%	37.53%	**38.11**%
R101-C4 mAP all	30.17%	36.7%	37.2%	37.4%	**38.86**%	38.23%
R101-FPN mAP all	31.67%	38.6%	38.8%	39.3%	40.25%	**41.7**%
R101-FPN mAP s	14.25%	17.37%	18.18%	18.35%	19.33%	**22.24**%
R101-FPN mAP m	37.53%	39.23%	39.74%	40.52%	41.27%	**43.26**%
R101-FPN mAP l	50.14%	51.64%	51.83%	52.17%	52.22%	**53.27**%

4.4.3 Implications of my finding regarding the wave loss function

Our results clearly demonstrate that incorporating topological information in the loss function can improve the segmentation accuracy of various network architectures. The results in Table 4.1 and 4.2 demonstrate that our approach not only improved the accuracy by 3% on average but also performs better than any of the other loss functions selected for comparison. One can easily see that incorporating more topographic information in the loss functions improved segmentation accuracy and the best results could be achieved with wave loss. We also have to note that our method is not another completely different loss function, but a combination and generalization of area- and distance-based metrics. For example, setting the $SpaW$ parameter to an all zero vector except the first value will result in the spatial information being incorporated only at one pixel from the intersection, which is exactly the same region as the boundary of the object. This way, one can calculate boundary loss using our method. Similarly, if all $SpaW$ values are zero our metric will compute an area-based metric, similar to the Hamming distance or Dice score. On the other hand, if $ValW$ parameters

are all set to zero, only the distance of the differing pixels will be a determining factor similarly to Hausdorff distance. These results show that our approach defines a more general metric which can mimic most of the previously applied loss functions and with proper parameterization it can also perform better in practical applications.

4.5 Conclusions

In this thesis, I have shown how a topographic metric can help in the increase of the accuracy of commonly applied image segmentation networks during training, and results in higher accuracy and precision in evaluation. We have shown on a simple dataset, inspired by CLEVR, that the same network can achieve better accuracy and faster convergence using wave loss rather than pixel-based loss functions.

We have also shown, in more complex tasks, that the overall accuracy of instance segmentation could be increased by 3% on MS-COCO using the Mask R-CNN architecture, with a ResNet-101 backbone, modifying only the loss function from cross entropy to wave loss.

We have also demonstrated on the Cityscapes dataset that the inclusion of topographic information in the loss function can increase the test accuracy by 3% on average compared to cross entropy, which was observed in the case of four different architectures (SegNet, DeepLab, DeepLabV3 and HRNet). We also compared wave loss to other recently published loss functions such as Dice loss, active boundary loss and shape aware loss and our approach provided higher segmentation accuracy in all cases.

These results are initial and further detailed investigations are needed using various networks, datasets and parameter settings, but we believe they are promising and demonstrate that including topographic information in loss calculation can result in higher IOU measures in all segmentation problems.

Chapter 5

Discussion

The robustness test and subsequent clinical pre-testing conducted with the involvement of esteemed physicians were pivotal in validating the efficacy and reliability of the developed denoising tool. Through rigorous testing across various scenarios and datasets, the tool showcased consistent performance, effectively mitigating noise in bone scintigrapy while preserving crucial diagnostic information. Feedback from clinicians has highlighted the device's ability to improve image clarity, speeding up diagnosis and increasing physician confidence.

The involvement of physicians in the pre-testing phase provided invaluable insights into the practical utility of the denoising tool. Their positive feedback emphasized the ease of integration into existing clinical workflows and its tangible impact on diagnostic accuracy. Physicians noted a marked improvement in image quality, enabling clearer identification of anatomical structures and pathological features. This user-centered approach ensured that the developed solution not only met technical benchmarks but also addressed the pressing needs and preferences of end-users in the clinical setting.

5.1 Utilization as a Fast Localizer

The uniqueness of the method lies in its adaptability across diverse bone scintigrapy imaging scenarios while maintaining a high level of denoising efficacy. The first application of the developed denoising method lies in its remarkable capability as a fast localizer for rapid measurements in SPECT imaging. The method's

efficacy in swiftly processing extremely noisy images allows for quick preliminary assessments and localization of regions of interest. This functionality is particularly advantageous in time-sensitive scenarios, enabling prompt identification or initial localization of anatomical structures or abnormalities.

It's essential to acknowledge that the denoising method, while proficient in improving the quality of rapid measurements, does not render these images suitable for diagnostic purposes. The nature of these fast measurements inherently produces highly noisy images that lack the requisite level of detail and fidelity necessary for accurate clinical diagnosis. As such, caution must be exercised in interpreting these denoised images, emphasizing that they serve as rapid localizers rather than diagnostically conclusive representations. Additionally, studying the correlation between denoised localizer images and subsequent detailed diagnostic images could provide valuable insights into the method's predictive capabilities.

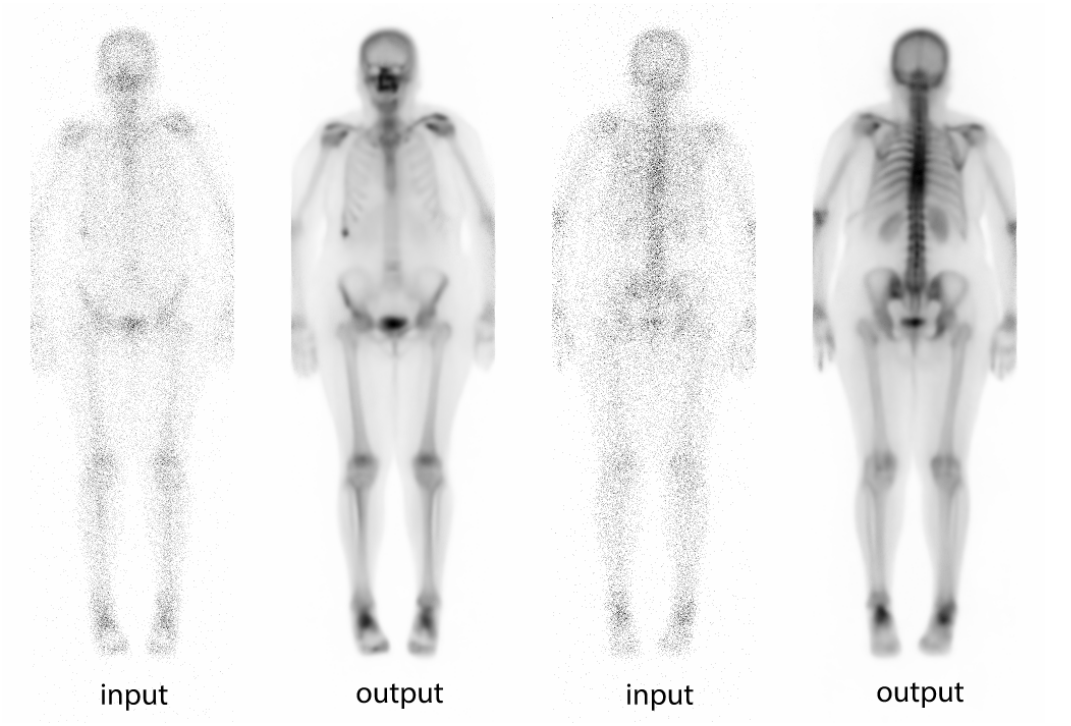


Figure 5.1: Fast localizer sample image (≈ 40 sec) taken during patient start positioning.

5.2 Utilization as a diagnostic noise filter

In considering the future prospects of the developed denoising method, a compelling perspective emerges when contemplating the possibility of leveraging a rigorous clinical validation process to potentially transition towards utilizing this method for standard diagnostic measurements. Upon successful validation, the pathway forward holds promise in reshaping diagnostic practices by advocating for lower radiation doses and shorter measurement times while maintaining or surpassing the current standard of diagnostic reliability. The envisioned evolution entails conducting comprehensive clinical studies to establish the method's diagnostic accuracy, reliability, and safety in diverse patient cohorts and clinical settings. If validated rigorously, this approach could revolutionize bone scintigraphy imaging by offering a method that not only mitigates radiation exposure for patients but also significantly reduces acquisition times without compromising diagnostic confidence. This prospect not only underscores the method's potential but also heralds a transformative shift towards more efficient, patient-friendly, yet equally, if not more, reliable diagnostic protocols in clinical practice.

5.3 The development of a physician-assisting diagnostic solution

The ongoing focus of our team's endeavors centers on an advanced solution that extends beyond mere noise filtering. Our efforts have culminated in a multifaceted approach that integrates sophisticated segmentation algorithms to delineate anatomical structures within the images. This intricate segmentation process serves as a foundation for a subsequent classification framework, enabling the identification and differentiation of distinct accumulations within the image. These accumulations are categorized into three groups: benign, malignant, and artifact types. By employing machine learning and pattern recognition techniques, our solution accurately discerns and categorizes these clusters based on their distinct characteristics and features. Consequently, the output presented to the physician is not merely a denoised image but a structured artifact highlighting these identified clusters, serving as an invaluable starting point for further detailed

analysis and clinical interpretation. This approach aims to assist clinicians by offering a structured and comprehensive report, facilitating expedited and more informed decision-making processes in the diagnosis and evaluation of SPECT imaging results.

Chapter 6

Summary

I have shown in my thesis that deep learning can be effectively utilized for high-quality and reliable noise filtering in planar bone scintigraphy. The proposed solution for deep learning-based noise filtering for planar bone scintigraphy is robust and I have established a reliable evaluation method without the need for noise-free, perfect images for comparison. I have also presented that it is possible to develop a loss function for deep learning that considers the topographical structure of segmentations, as opposed to just pixel-level comparisons, to improve the accuracy of the noise filtering tool in planar bone scintigraphy.

I have developed a robust high-quality noise filtering method tailored for planar bone scintigraphy, showcasing the effectiveness of deep learning in this context. Through validation on a real patient database, isolated for accuracy assessment, the top-performing neural network achieved a mean RMSE of 1.15 under normal statistics, outperforming the best non-neural network solution, BM3D, which attained a mean error of 1.29. Furthermore, when tested under 1/3 statistics, the neural network yielded an average RMSE of 1.38, surpassing the best non-neural network-based solution, Gaussian 9mm, which attained an RMSE of 2.07. These findings unequivocally demonstrate the superior noise filtering capabilities of neural networks compared to established non-neural network methods used in clinical practice.

6.1 New scientific results

Thesis I a: *I have developed a robust high-quality noise filtering method tailored for planar bone scintigraphy, showcasing the effectiveness of deep learning in this context. Through validation on a real patient database, isolated for accuracy assessment, the top-performing neural network achieved a mean RMSE of 1.15 under normal statistics, outperforming the best non-neural network solution, BM3D, which attained a mean error of 1.29. Furthermore, when tested under 1/3 statistics, the neural network yielded an average RMSE of 1.38, surpassing the best non-neural network-based solution, Gaussian 9mm, which attained an RMSE of 2.07. These findings demonstrate the superior noise filtering capabilities of neural networks compared to established non-neural network methods used in clinical practice.* Corresponding publication: [83]

In this thesis a neural network based noise filter is proposed that can be used with planar bone scintigraphy recordings at multiple noise levels, instead of developing a separate network for each noise level. The proposed denoising solution is a convolutional neural network (CNN) inspired by U-NET architecture. A total of 1215 pairs of anterior and posterior patient images were available for training and evaluation during the analysis. The noise-filtering network was trained using bone scintigraphy recordings with real statistics according to the standard protocol, without noise-free recordings. The resulting solution proved to be robust to the noise level of the images within the examined limits. During the evaluation, the performance of the networks was compared to Gaussian and median filters and to the Block-matching and 3D filtering (BM3D) filter. It has been shown that particularly high signal-to-noise ratios can be achieved using noise-filtering neural networks (NNs), which are more robust than the traditional methods and can help diagnosis, especially for images with high noise content.

From Table 3.2 showing the results by RMSE metric, it can be seen that for all statistics, the neural network based solutions achieved the best results. Note that under normal and 1/3 statistics, at this metric, the performance of the BM3D and Gaussian filters is comparable to the neural network, but with worse statistics, the performance of these solutions degrades to unusable levels.

Thesis I b: *My deep learning-based noise filtering solution for planar bone scintigraphy demonstrates robustness in real-life applications. Through a comprehensive investigation across various homogeneous and biased validation datasets (including diverse age groups, BMI ranges, and gender categories), I assessed the performance variability of the denoising algorithm. The evaluation revealed notable consistency and effectiveness, showcasing that for datasets with normal statistics, the average RMSE error ranged from 1.05 to 1.28 across different subsets, with the mixed dataset registering an average error of 1.15. Moreover, the standard deviation within each subset ranged from 0.4 to 0.48, highlighting the stability and reliability of the filtering algorithm. When evaluated under 1/3 statistics, the mean error exhibited a similar range, varying from 1.28 to 1.54, with standard deviations ranging from 0.41 to 0.49. These findings underscore the adaptability and consistent performance of our deep learning-based filter across diverse patient groups, reinforcing its robustness and applicability in real-world scenarios of planar bone scintigraphy imaging.* Corresponding publication: [83]

The measurement results on which the thesis claims are based are given in Table 3.5. The trends in performance measured on the different sets as a function of the deterioration of the statistics are the same as those observed on the mixed set.

Thesis I c: *I have created an effective evaluation method for the deep learning-driven noise filtering tool in planar bone scintigraphy without the need for noise-free images as a reference. In the advanced stages of development, we identified a neural network exhibiting satisfactory performance in processing low noise content measurements. Using this selected neural network, we created a noise-free validation dataset of 544 measurements and then had these images analyzed by physicians to identify any abnormalities, unusual structures, accumulations or artifacts compared to the original images. Within our evaluation framework, these filtered images were considered as virtually noise-free, representing idealized images. We further employed these "noise-free" images, under normal statistical conditions, to generate Poisson noise-affected images, serving as inputs for our solutions. Additionally, leveraging these artificially created standard measurement like images, I have applied additional degradation by employing binomial sam-*

pling, thus creating lower-quality representations for comparative assessments. Corresponding publication: [83]

The whole pipeline and the examples of the images produced by the pipeline are shown in Figure 3.5 and Figure 3.6.

Thesis II: *I have developed a training loss function for neural networks that considers the topographical structure of segmentations, as opposed to just pixel-level comparisons. The proposed method has increased segmentation accuracy by* 3% *on both the Cityscapes and MS-COCO datasets compared to cross entropy, using various network architectures.* Corresponding publication: [84]

The solution of segmentation problems with deep neural networks requires a well-defined loss function for comparison and network training. In most network training approaches, only area-based differences that are of differing pixel matter are considered; the distribution is not. Our brain can compare complex objects with ease and considers both pixel level and topological differences simultaneously. Comparison between objects requires a properly defined metric that determines similarity between them considering changes both in shape and values. In past years, topographic aspects were incorporated in loss functions where either boundary pixels or the ratio of the areas were employed in difference calculation. During our work we showed how the application of a topographic metric, called wave loss, can be applied in neural network training and increase the accuracy of traditional segmentation algorithms. The proposed method has increased segmentation accuracy by 3% on both the Cityscapes and MS-COCO datasets compared to cross entropy, using various network architectures.

Journal publications of the thesis

[41] Á. Kovács, G. Légrádi, A. Wirth, F. Nagy, A. Forgács, S. Barna, I. Garai, and T. Bükki, "A mesterséges és emberi intelligencia értéke a csontszcintigráfia példáján keresztül," Magyar Onkológia, vol. 64, no. 2, pp. 153–158, 2020

[83] A. Kovacs, T. Bukki, G. Legradi, N. J. Meszaros, G. Z. Kovacs, P. Prajczer, I. Tamaga, Z. Seress, G. Kiszler, A. Forgacs, S. Barna, I. Garai, and A. Hor- vath, "Robustness analysis of denoising neural networks for bone scintigraphy," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 1039, p. 167003, sep 2022.

[84] Á. Kovács, J. Al-Afandi, C. Botos, and A. Horváth, "Wave Loss: A Topographic Metric for Image Segmentation," Mathematics 2022, Vol. 10, Page 1932, vol. 10, p. 1932, jun 2022.

Acknowledgement

I would like to extend my heartfelt thanks to the numerous individuals and organizations who have supported me throughout my PhD journey. Their contributions have been invaluable in helping me reach this important milestone.

First, I would like to express my gratitude to my company (Mediso Ltd.) for their unwavering support and encouragement throughout my studies. Their investment in my education and research has been fundamental to my success, and I am deeply grateful for the opportunities and resources they have provided.

I am also fortunate to have had the guidance and support of two exceptional advisors. Their expertise, wisdom, and mentorship have been instrumental in shaping my research and developing my skills as a scholar. I am grateful for the countless hours they have spent guiding and supporting me, and for the profound impact they have had on my life and work.

My doctoral school has also played a crucial role in my success. Their commitment to academic excellence, the resources and infrastructure they have provided, and the support they have given throughout my PhD journey have been greatly appreciated.

Finally, I would like to thank my family for their unwavering love, support, and encouragement. Their belief in me has been a constant source of inspiration, and I am grateful for their constant love and support throughout this journey.

I am deeply grateful to all those who have supported me on this journey, and I hope that this research will make a meaningful contribution to the field.

References

[1] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning image restoration without clean data," in *35th International Conference on Machine Learning, ICML 2018*, vol. 7, pp. 4620–4631, International Machine Learning Society (IMLS), 2018. 1.1, 2.2, 3

[2] E. Bercovich and M. C. Javitt, "Medical Imaging: From Roentgen to the Digital Revolution, and Beyond," *Rambam Maimonides Medical Journal*, vol. 9, p. e0034, oct 2018. 2.1

[3] E. J. van Beek and E. A. Hoffman, "Functional Imaging: CT and MRI," *Clinics in chest medicine*, vol. 29, p. 195, mar 2008. 2.1

[4] M. N. Wernick and J. N. Aarsvold, *Emission Tomography: The Fundamentals of PET and SPECT*. Elsevier Academic Press, 2004. 2.1, 2.2

[5] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A Survey on Deep Learning in Medical Image Analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, feb 2017. 2.2

[6] D. Shen, G. Wu, and H. I. Suk, "Deep Learning in Medical Image Analysis," *Annual Review of Biomedical Engineering*, vol. 19, pp. 221–248, jun 2017. 2.2

[7] M. I. Razzak, S. Naz, and A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," in *Lecture Notes in Computational Vision and Biomechanics*, vol. 26, pp. 323–350, Springer Netherlands, 2018. 2.2

[8] C. Han, L. Rundo, K. Murao, T. Nemoto, and H. Nakayama, "Bridging the gap between AI and Healthcare sides: towards developing clinically relevant AI-powered diagnosis systems," *IFIP Advances in Information and Communication Technology*, vol. 584 IFIP, pp. 320–333, jan 2020. 2.2

[9] M. Elhoseny and K. Shankar, "Optimal bilateral filter and Convolutional Neural Network based denoising method of medical image measurements," *Measurement: Journal of the International Measurement Confederation*, vol. 143, pp. 125–135, sep 2019. 2.2

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning* (E. R. Dougherty, J. T. Astola, K. O. Egiazarian, N. M. Nasrabadi, and S. A. Rizvi, eds.), vol. 6064, p. 606414, SPIE, feb 2006. 2.2.1, 2.2.2

[11] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," 2019. 2.2.2

[12] M. Mäkitalo and A. Foi, "On the inversion of the anscombe transformation in low-count poisson image denoising," in *2009 International Workshop on Local and Non-Local Approximation in Image Processing, LNLA 2009*, pp. 26–32, 2009. 2.2.2

[13] A. Foi, "Image and video denoising by sparse 3D transformdomain collaborative filtering," *Transforms and spectral methods group, Department of signal processing, Tampere university, http://www. cs. tut. fi/foi/GCF-BM3D/, accessed Aug*, vol. 3, 2014. 2.2.2, 3.4.1

[14] M. Mäkitalo and A. Foi, "Optimal inversion of the generalized anscombe transformation for Poisson-Gaussian noise," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 91–103, 2013. 2.2.2, 2.2.2, 2.2.2, 2.2.2

[15] H. Choi, "Deep Learning in Nuclear Medicine and Molecular Imaging: Current Perspectives and Future Directions," *Nuclear Medicine and Molecular Imaging*, vol. 52, p. 109, apr 2018. 2.3

[16] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, pp. 1–27, jan 2009. 2.3

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, dec 2015. 2.3

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012. 2.3, 2.3.1

[19] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature 2015 521:7553*, vol. 521, pp. 436–444, may 2015. 2.3

[20] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nature Biotechnology 2015 33:8*, vol. 33, pp. 831–838, jul 2015. 2.3

[21] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning–based sequence model," *Nature Methods 2015 12:10*, vol. 12, pp. 931–934, aug 2015. 2.3

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature 1986 323:6088*, vol. 323, no. 6088, pp. 533–536, 1986. 2.3

[23] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems*, pp. 153–160, 2007. 2.3

[24] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504–507, jul 2006. 2.3

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 1–9, oct 2015. 2.3.1

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016-December, pp. 770–778, jun 2016. 2.3.1

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, Springer Verlag, 2015. 2.3.2, 3.2, 3.3.2, 3.3.2

[28] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3234–3243, 2016. 2.3.2

[29] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, dec 2017. 2.3.2, 4.4.1

[30] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017. 2.3.2

[31] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 2999–3007, dec 2017. 2.3.2

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 431–440, oct 2015. 2.3.2

[33] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence*

and Lecture Notes in Bioinformatics), vol. 8695 LNCS, no. PART 7, pp. 345–360, 2014. 2.3.2

[34] Y. Zhu, Y. Tian, D. Metaxas, and P. Dollar, "Semantic amodal segmentation," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3001–3009, nov 2017. 2.3.2

[35] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for L1 regularization: A comparative study and two new approaches," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4701 LNAI, pp. 286–297, 2007. 2.3.2

[36] X. Hu, L. Fuxin, D. Samaras, and C. Chen, "Topology-preserving deep image segmentation," in *Advances in Neural Information Processing Systems*, vol. 32, 2019. 2.3.2

[37] J. Clough, N. Byrne, I. Oksuz, V. A. Zimmer, J. A. Schnabel, and A. King, "A Topological Loss Function for Deep-Learning based Image Segmentation using Persistent Homology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, oct 2020. 2.3.2

[38] S. Shit, J. C. Paetzold, A. Sekuboyina, I. Ezhov, A. Unger, A. Zhylka, J. P. Pluim, U. Bauer, and B. H. Menze, "CLDICE - A novel topology-preserving loss function for tubular structure segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 16555–16564, 2021. 2.3.2

[39] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. Ben Ayed, "Boundary loss for highly unbalanced segmentation," *Medical Image Analysis*, vol. 67, p. 101851, jan 2021. 2.3.2

[40] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, Association for Computing Machinery (ACM), 2008. 3

[41] Á. Kovács, G. Légrádi, A. Wirth, F. Nagy, A. Forgács, S. Barna, I. Garai, and T. Bükki, "A mesterséges és emberi intelligencia értéke a csontszcintigráfia példáján keresztül," *Magyar Onkológia*, vol. 64, no. 2, pp. 153–158, 2020. 3, 3.5, 3.4.3, 6.1

[42] N. Yuan, J. Zhou, and J. Qi, "Half2Half: deep neural network based CT image denoising without independent reference data," *Physics in Medicine and Biology*, vol. 65, p. 215020, nov 2020. 3, 3.1

[43] I. A. Elbakri and J. A. Fessier, "Segmentation-free statistical image reconstruction for polyenergetic X-ray computed tomography," in *Proceedings - International Symposium on Biomedical Imaging*, vol. 2002-Janua, pp. 828–831, 2002. 3

[44] G. Last and M. Penrose, *Lectures on the Poisson Process*. No. August in Institute of Mathematical Statistics Textbooks (7), Cambridge University Press, 2017. 3

[45] C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer International Publishing, 2018. 3.1, 3.2

[46] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010. 3.2

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. 3.2

[48] F. Nagy, A. K. Krizsan, K. Kukuts, M. Szolikova, Z. Hascsi, S. Barna, A. Acs, P. Szabo, L. Tron, L. Balkay, M. Dahlbom, M. Zentai, A. Forgacs, and I. Garai, "Q-Bot: automatic DICOM metadata monitoring for the next level of quality management in nuclear medicine," *EJNMMI Physics*, vol. 8, pp. 1–13, dec 2021. 3.3.1

[49] J. Teuwen and N. Moriakov, "Convolutional neural networks," in *Handbook of Medical Image Computing and Computer Assisted Intervention*, pp. 481–501, Academic Press, jan 2019. 3.3.2, 3.3.2

[50] F. Chollet and Others, "Keras." `https://keras.io`, 2015. 3.3.2

[51] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, pp. 265–283, 2016. 3.3.2, 19

[52] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, dec 2014. 3.3.2

[53] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7700 LECTU, pp. 9–48, 2012. 3.3.2

[54] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings - International Conference on Pattern Recognition*, pp. 2366–2369, 2010. 3.4

[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, apr 2004. 3.4

[56] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and T. scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 2014. 3.4

[57] Y. Hou, C. Zhao, D. Yang, and Y. Cheng, "Comments on image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 20, pp. 268–270, jan 2011. 3.4.1

[58] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, pp. 2080–2095, aug 2007. 3.4.1

[59] D. Minarik, O. Enqvist, and E. Trägårdh, "Denoising of scintillation camera images using a deep convolutional neural network: A Monte Carlo simulation approach," *Journal of Nuclear Medicine*, vol. 61, no. 2, pp. 298–303, 2020. 3.4.1

[60] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978. 3.4.2, 3.7, 3.10

[61] Plotly Technologies Inc., "Box plots in Python," 2022. 3.4.2, 3.7, 3.10

[62] R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950. 4.1

[63] J. Henrikson, "Completeness and total boundedness of the Hausdorff metric," *MIT Undergraduate Journal of Math.*, vol. 1, pp. 69–79, 1999. 4.1, 4.1

[64] R. Zhao, B. Qian, X. Zhang, Y. Li, R. Wei, Y. Liu, and Y. Pan, "Rethinking dice loss for medical image segmentation," *Proceedings - IEEE International Conference on Data Mining, ICDM*, vol. 2020-November, pp. 851–860, nov 2020. 4.1, 4.3.2

[65] M. Berman, A. R. Triki, and M. B. Blaschko, "The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4413–4421, dec 2018. 4.1

[66] N. Abraham and N. M. Khan, "A novel focal tversky loss function with improved attention u-net for lesion segmentation," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2019-April, pp. 683–687, apr 2019. 4.1

[67] S. I. Outcalt and M. A. Melton, "Geomorphic application of the hausdorff-besicovich dimension," *Earth Surface Processes and Landforms*, vol. 17, pp. 775–787, dec 1992. 4.1

[68] F. Latrémolière, "The quantum Gromov-Hausdorff propinquity," *Transactions of the American Mathematical Society*, vol. 368, pp. 365–411, may 2016. 4.1

[69] I. Szatmári, C. Rekeczky, and T. Roska, "A Nonlinear Wave Metric and its CNN Implementation for Object Classification," *Journal of VLSI signal processing systems for signal, image and video technology 1999 23:2*, vol. 23, pp. 437–447, nov 1999. 4.1

[70] T. Roska and L. Chua, "The CNN Universal Machine: An Analogic Array Computer," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 3, pp. 163–173, 1993. 4.1

[71] J. Al-Afandi and A. Horvath, "Application of the Nonlinear Wave Metric for Image Segmentation in Neural Networks," in *CNNA 2018; The 16th International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1–4, 2018. 4.1

[72] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, and ..., *Automatic differentiation in pytorch*. 2017. 19, 4.3

[73] J. Johnson, L. Fei-Fei, B. Hariharan, C. L. Zitnick, L. Van Der Maaten, and R. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1988–1997, Institute of Electrical and Electronics Engineers Inc., nov 2017. 4.3, 4.3.1

[74] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 3213–3223, IEEE Computer Society, dec 2016. 4.3, 4.4.1

[75] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014. 4.3, 4.4.2

[76] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019. 4.3

[77] Y. LeCun, C. Cortes, and C. J. Burges, "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges," 1998. 4.3.1

[78] C. Wang, Y. Zhang, M. Cui, P. Ren, Y. Yang, X. Xie, X.-S. Hua, H. Bao, and W. Xu, "Active Boundary Loss for Semantic Segmentation," feb 2021. 4.3.2

[79] S. M. R. Al Arif, K. Knapp, and G. Slabaugh, "Shape-aware deep convolutional neural network for vertebrae segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10734 LNCS, pp. 12–24, Springer Verlag, 2018. 4.3.2

[80] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "HigherhrNet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 5385–5394, 2020. 4.4.1

[81] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," *Arxiv*, pp. 1–12, dec 2014. 4.4.1

[82] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, apr 2018. 4.4.1

[83] A. Kovacs, T. Bukki, G. Legradi, N. J. Meszaros, G. Z. Kovacs, P. Prajczer, I. Tamaga, Z. Seress, G. Kiszler, A. Forgacs, S. Barna, I. Garai, and A. Horvath, "Robustness analysis of denoising neural networks for bone scintigraphy," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1039, p. 167003, sep 2022. 6.1

[84] Á. Kovács, J. Al-Afandi, C. Botos, and A. Horváth, "Wave Loss: A Topographic Metric for Image Segmentation," *Mathematics 2022, Vol. 10, Page 1932*, vol. 10, p. 1932, jun 2022. 6.1