Nuclear Inst. and Methods in Physics Research, A 1039 (2022) 167003

Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

# Robustness analysis of denoising neural networks for bone scintigraphy

Akos Kovacs [a,d,*], Tamas Bukki [a], Gabor Legradi [a], Nora J. Meszaros [a], Gyula Z. Kovacs [a], Peter Prajczer [a], Istvan Tamaga [a], Zoltan Seress [a], Gabor Kiszler [a], Attila Forgacs [a,b], Sandor Barna [b], Ildiko Garai [b,c], Andras Horvath [d]

[a] Mediso Ltd., Laborc utca 3., Budapest, 1037, Hungary
[b] ScanoMed Nuclear Medicine Centers, Nagyerdei krt. 98, Debrecen, 4032, Hungary
[c] Division of Nuclear Medicine, Department of Medical Imaging, Faculty of Medicine, University of Debrecen, Egyetem ter 1, Debrecen, 4032, Hungary
[d] Faculty of Information Technology and Bionics, Pazmany Peter Catholic University, Prater utca. 50/A, Budapest, 1083, Hungary

ARTICLE INFO

*Keywords:*
Image denoising
Poisson noise
Machine learning
Convolutional neural network
Deep learning
Bone scintigraphy
Medical imaging

ABSTRACT

This paper describes and compares two neural network (NN) based noise filters developed for planar bone scintigraphy.

Images taken with a gamma camera typically have a low signal-to-noise ratio and are subject to significant Poisson noise. In our work, we have designed a neural network based noise filter 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, we compared the performance of the networks to Gaussian and median filters and to the Block-matching and 3D filtering (BM3D) filter. Our presented evaluation method in this article does not require noiseless images and we measured the performance and robustness of our solution on specialized validation sets.

We showed 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.

## 1. Introduction

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.

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 specialized 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 [1].

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 [2]. 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),

* Corresponding author at: Mediso Ltd., Laborc utca 3., Budapest, 1037, Hungary.
*E-mail address:* akos.kovacs@mediso.com (A. Kovacs).

https://doi.org/10.1016/j.nima.2022.167003
Received 7 March 2022; Received in revised form 19 May 2022; Accepted 31 May 2022
Available online 20 June 2022

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.

In the following, we investigate the performance and robustness of our NN-based noise filter on bone scintigraphy images. However, the methodology presented here can be generalized to investigate the robustness of other ML algorithms used in medical image processing.

### 1.1. Characteristics of gamma camera bone scintigraphy

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 [3].

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.

### 1.2. 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:

$$y \approx f(\boldsymbol{x}) \tag{1}$$

The $f$ used to cover traditional noise filtering algorithms such as Gaussian or median filtering, or possibly more complicated methods such as BM3D [4]. In this paper, 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.

### 1.3. Traditional noise filtering solutions for bone scintigraphy

In addition to the commonly used Gaussian and median filtering in the literature [5], 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 [4].

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 [6]. The BM3D implementation used in our paper is available at [7].

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 our paper. Instead of pure Anscombe transformation, we achieved better image quality by using the generalized Anscombe transformation [8], see Eq. (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 [8], see Eqs. (3) and (4).

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

Closed form approximation of unbiased inverse Anscombe transformation [8]:

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

Closed form approximation of unbiased generalized inverse Anscombe transformation [8], where $A^{-1}$ comes from Eq. (3):

$$A^{-1} : x \mapsto \begin{cases} A^{-1}(x)\alpha + \mu, & A^{-1}(x) > 0 \\ \mu, & A^{-1}(x) < 0 \end{cases} \tag{4}$$

In Eqs. (2), (3) and (4) $\overset{*}{z}$ is the observed pixel value obtained through an image acquisition device. In Ref. [8] 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 $\mu$ and standard deviation $\overset{*}{\sigma}$.

## 2. Noise filtering of high-noise planar bone scintigraphy images using deep learning

### 2.1. Dataset creation process

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. [9]. 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 [2]. 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 [10].

According to the article [11], one possible strategy is to put an artificially degraded, noisy image on the input side and keep the full statistics 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. 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 modeled by the sum of a Poisson distribution and a Gaussian distribution [12].

According to Ref. [13], 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 $Z_1$, $Z_2$ … 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

$$X := \sum_{j=1}^{Z} Z_j \tag{5}$$

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.

### 2.2. 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 [11] 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. 1), but it was less effective at filtering noise in several regions of the image (Fig. 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 localization 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 [14].

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.

### 2.3. Neural network architecture

The authors of Ref. [15] 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 [16], which has achieved outstanding results on ImageNet.

For the sake of simplicity, this article is limited to introducing a solution based on the famous U-NET architecture [17].

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 [17].

This artificial neural network combines different layers of convolution [14] and max-pooling [14] see Fig. 3. The peculiarity of this network lies in its skip-connections, whereby lower layers receive textural information from higher 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 [17].

## 3. Materials and methods

### 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 [18]. 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™ processing SW (Mediso Ltd.). All patients were given 5–600MBq Tc-99 m methylene diphosphonate (MDP) (Isotope Institute LTD, Budapest, Hungary) intravenously with 2–5 h accumulation time. The matrix size was 256*1024 with 130 mm/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.

### 3.2. Detailed description of neural network architectures

In this article, we present the results of two neural networks with U-NET architecture [17], 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. (6)) in the networks [19].

$$\mathrm{ReLU}(x) = max(0, x), x \in (\mathbb{R}) \tag{6}$$

The network builds from an encoder branch, which compresses the information, and a decoder branch, which reconstructs the image. The

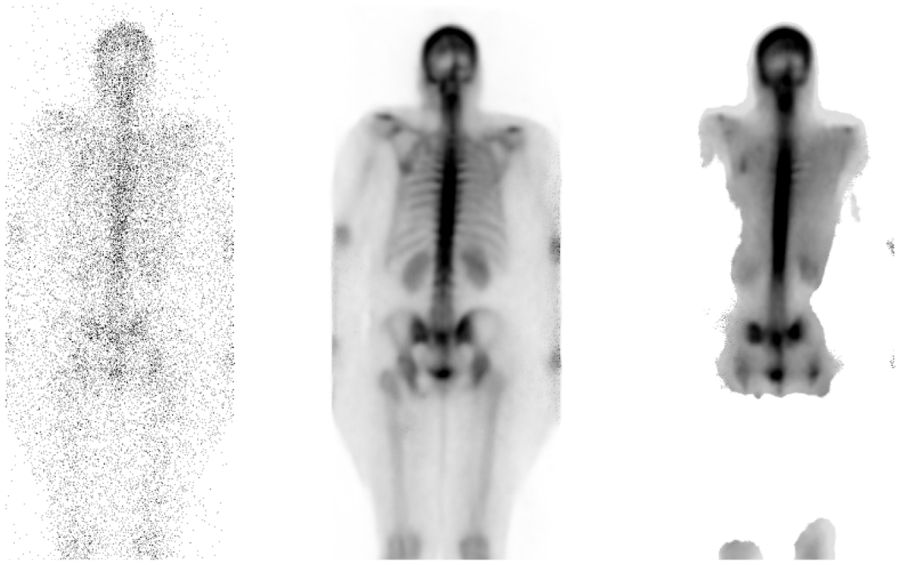



**Fig. 1.** With greatly reduced statistics (1/32) compared to normal, the network trained with the MAE loss function tends to cut out or zero out low-impact regions during filtering. The figure shows a very strong version of this phenomenon.

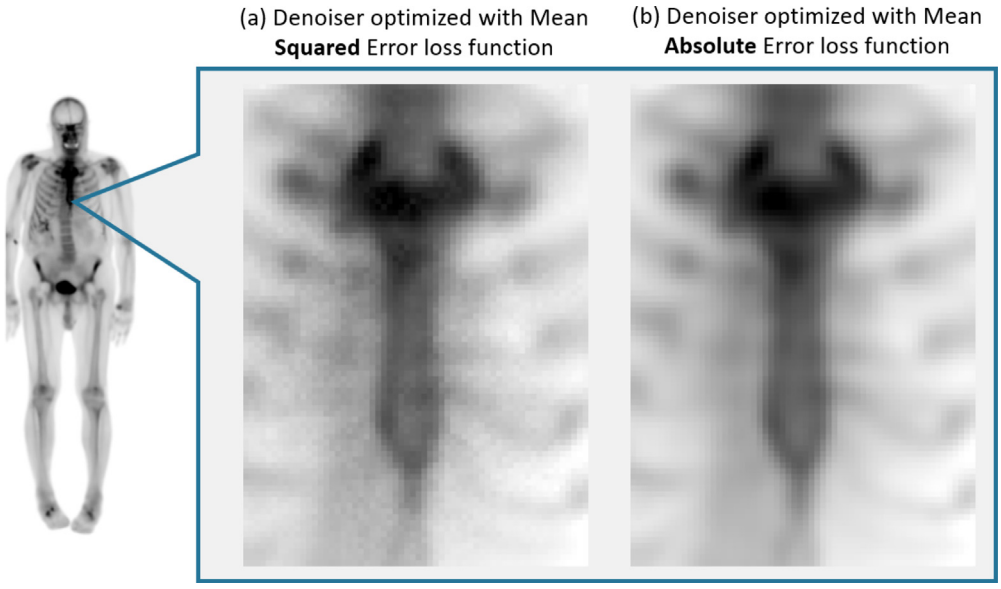



**Fig. 2.** Choosing MSE as the optimization cost function, (a) the network tended to be less effective at filtering, while the usage of MAE (b) results in a smoother image.

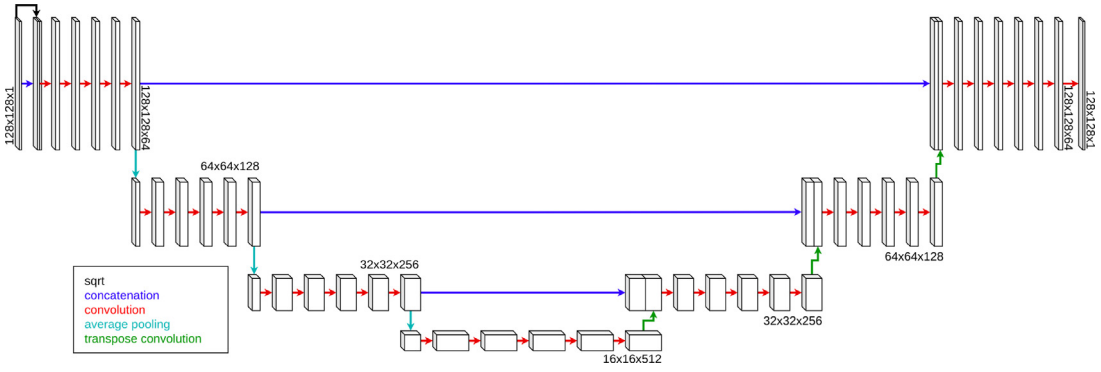



**Fig. 3.** The network architectures used for noise filtering were all U-NET based. The differences between the architectures were in the number of convolutional layers per level, and the number of convolutional filters. The network, named *L-NN*, contained 4 levels and always had 5 convolutional layers 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.

**Table 1**
Architecture details. 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.

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

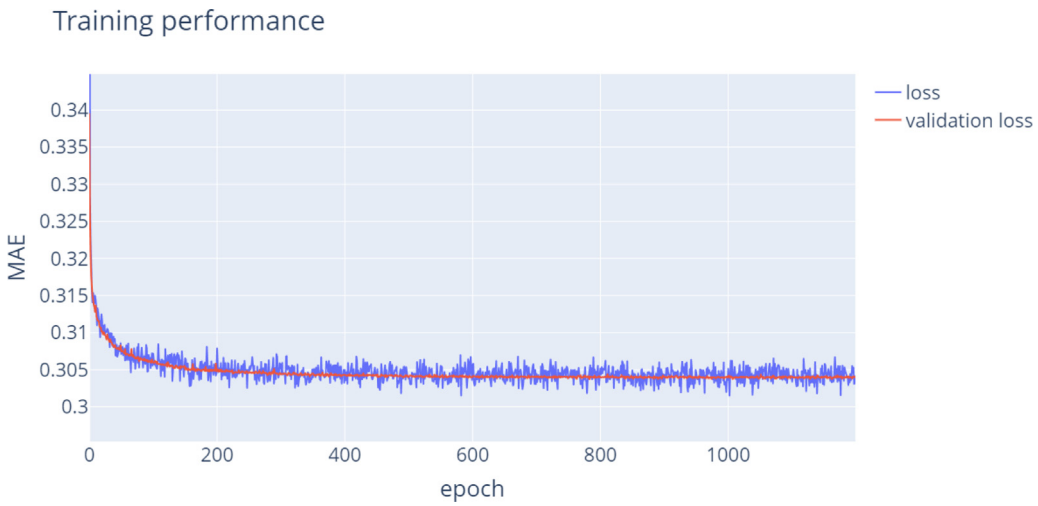



**Fig. 4.** The training performance of a neural network called S-NN. It can be seen that the error function is noisy due to the augmentation of the training data. Our validation data is static, where we can observe a slow improvement in performance.

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 [19]. After each level on 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 [17], 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 paper 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 [19].

A detailed summary of the architectures can be found in Table 1.

For training, we used NVIDIA GTX 1080 GPUs and relied on Keras [20] as ML software, with TensorFlow 2.4.1 backend library [21].

As optimizer we chose Adam [22] optimizer and we trained the networks with a learning rate of 0.001 for 1200 epochs. We did the training on $128 \times 128$ cropped patches instead of using the whole images for larger possible batch size [23], and also considered this method as an augmentation. In practical use, we run the neural network on the full 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. 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 paper 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

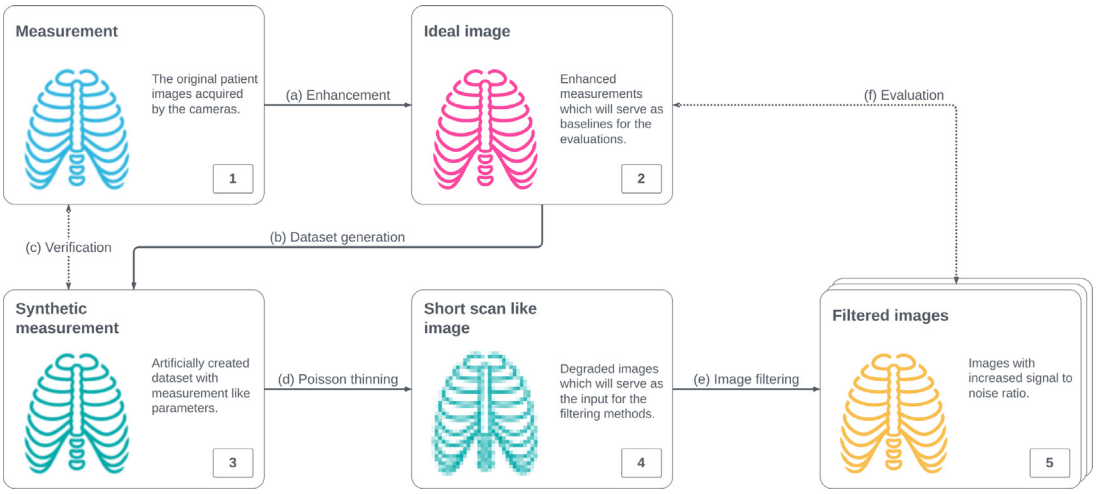



**Fig. 5.** Evaluation pipeline: We start from the real measurements acquired by the scanner (1). The second step is to create a noise-free image (2) with a reference enhancement solution (a), which was a neural network based denoiser in our case [10]. The ideal image will be then examined by physicians 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).

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.

## 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 Figs. 5 and 6. Using this process, we can also measure the peak signal-to-noise ratio (PSNR) [24] of the filtered images, which is shown in the figure examples. Given a reference image $y$ and a test image $x$, both of size $M$x$N$, PSNR in 2D is defined as Eq. (7).

$$\mathrm{PSNR}(x,y) = 10\log_{10}\frac{(\max_{j=1}^{n} y_j)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{7}$$

The filtered images were compared to the *ideal* images using Root Mean Square Error (RMSE) Eq. (8), and Structural Similarity Index Measure (SSIM) [25] Eq. (10) and MAE Eq. (9) 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.

$$\mathrm{RMSE}(x,y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{8}$$

$$\mathrm{MAE}(x,y) = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{9}$$

$$\mathrm{SSIM}(x,y) = \frac{1}{n}\sum_{i=1}^{n}\frac{(2\mu_{x_i}\mu_{y_i} + C_1) + (2\sigma_{x_i y_i} + C_2)}{(\mu_{x_i}^2 + \mu_{y_i}^2 + C_1)(\sigma_{x_i}^2 + \sigma_{y_i}^2 + C_2)} \tag{10}$$

In Eq. (10) $\mu_{x_i}$ and $\mu_{y_i}$ 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 $y_i$; $\sigma_{x_i}$, $\sigma_{y_i}$ 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) [26]. We used the default parameters of *skimage.metrics.structural_similarity*, so we used $7 \times 7$ 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.

### 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.

The BM3D implementation used in our paper is available at [7]. 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

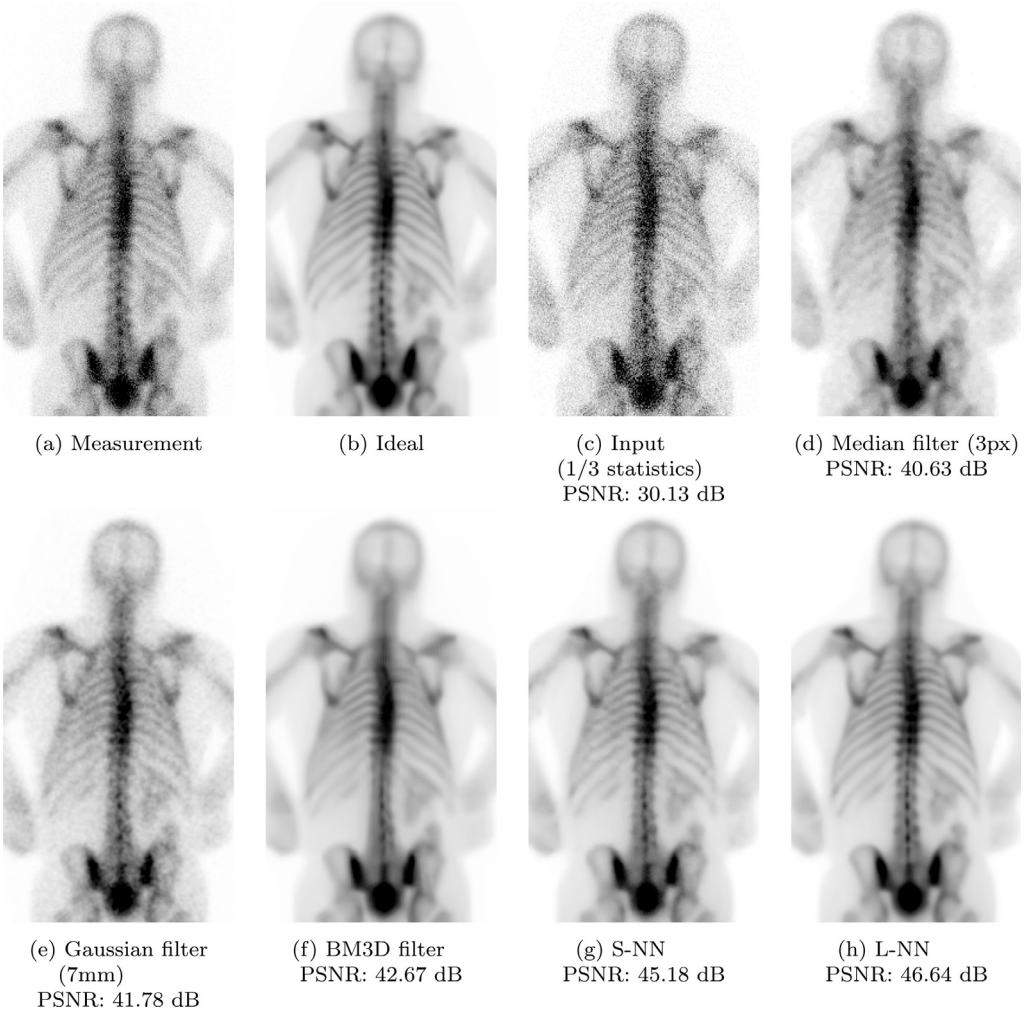



**Fig. 6.** Evaluation pipeline: We start from Measurement (a), from which we create a noise-free image with a reference filter (b). This will be then reviewed by doctors and taken as a benchmark. From this we generate an artificial degraded noisy image (c). Images (d), (e), (f), (g), (h) show the results of different filters. Since we have the noise-free reference image, we can correctly compute the errors of each method using the metrics.

authors of Ref. [7] 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 [27] as a better alternative to the profile originally proposed in [28] (which is currently the *vn_old* profile in the library), we still got better results with the previous preset. For the Generalized Anscombe transformation parameters (see Eq. (2), (3) and (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 Ref. [29]. In addition, we also measured four median filters with quadratic kernels of 9, 25, 49, and 81 pixels.

From Table 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. 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 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 Fig. 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. Fig. 8 with normal statistics and Fig. 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.

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

Based on the results shown in Tables 5 and 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

**Table 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.

| | 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 11 mm | 1.99 | 0.93 | 2.21 | 0.91 | 2.76 | 0.83 | 3.27 | 0.82 | 4.21 | 0.81 |
| Gaussian 13 mm | 2.34 | 1.21 | 2.48 | 1.19 | 2.85 | 1.12 | 3.23 | 1.10 | 3.94 | 1.05 |
| Gaussian 3 mm | 2.81 | 0.33 | 4.85 | 0.57 | 8.38 | 0.97 | 11.16 | 1.31 | 15.80 | 1.85 |
| Gaussian 5 mm | 1.69 | 0.31 | 2.66 | 0.37 | 4.44 | 0.53 | 5.87 | 0.70 | 8.26 | 0.97 |
| Gaussian 7 mm | 1.56 | 0.47 | 2.15 | 0.45 | 3.33 | 0.48 | 4.30 | 0.57 | 5.97 | 0.73 |
| Gaussian 9 mm | 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 3 px | 1.79 | 0.47 | 2.48 | 0.45 | 4.36 | 0.42 | 6.81 | 0.37 | 11.70 | 0.82 |
| Median 5 px | 2.64 | 0.91 | 3.06 | 0.87 | 4.50 | 0.72 | 6.71 | 0.55 | 11.51 | 0.82 |
| Median 7 px | 3.57 | 1.30 | 3.91 | 1.26 | 5.15 | 1.08 | 7.17 | 0.83 | 11.82 | 0.93 |
| Median 9 px | 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**
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 | |
| **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 11 mm | 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 13 mm | 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 3 mm | 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 5 mm | 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 7 mm | 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 9 mm | 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 5 px | 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 7 px | 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 9 px | 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 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 | |
| **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 11 mm | 1.12 | 0.35 | 1.32 | 0.33 | 1.78 | 0.33 | 2.18 | 0.34 | 2.88 | 0.39 |
| Gaussian 13 mm | 1.24 | 0.40 | 1.39 | 0.38 | 1.73 | 0.37 | 2.04 | 0.38 | 2.60 | 0.40 |
| Gaussian 3 mm | 2.00 | 0.22 | 3.43 | 0.37 | 5.87 | 0.63 | 7.81 | 0.84 | 10.94 | 1.25 |
| Gaussian 5 mm | 1.18 | 0.20 | 1.88 | 0.24 | 3.14 | 0.35 | 4.14 | 0.46 | 5.78 | 0.64 |
| Gaussian 7 mm | 1.03 | 0.25 | 1.47 | 0.25 | 2.33 | 0.30 | 3.03 | 0.36 | 4.20 | 0.48 |
| Gaussian 9 mm | 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 3 px | 1.16 | 0.28 | 1.74 | 0.26 | 3.22 | 0.27 | 5.26 | 0.36 | 9.40 | 0.81 |
| Median 5 px | 1.42 | 0.40 | 1.85 | 0.38 | 3.14 | 0.36 | 5.09 | 0.40 | 9.23 | 0.79 |
| Median 7 px | 1.77 | 0.52 | 2.17 | 0.52 | 3.39 | 0.47 | 5.27 | 0.45 | 9.38 | 0.81 |
| Median 9 px | 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 |

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 [30,31] computing MAE on different validation sets in Fig. 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.

## 5. Discussion

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

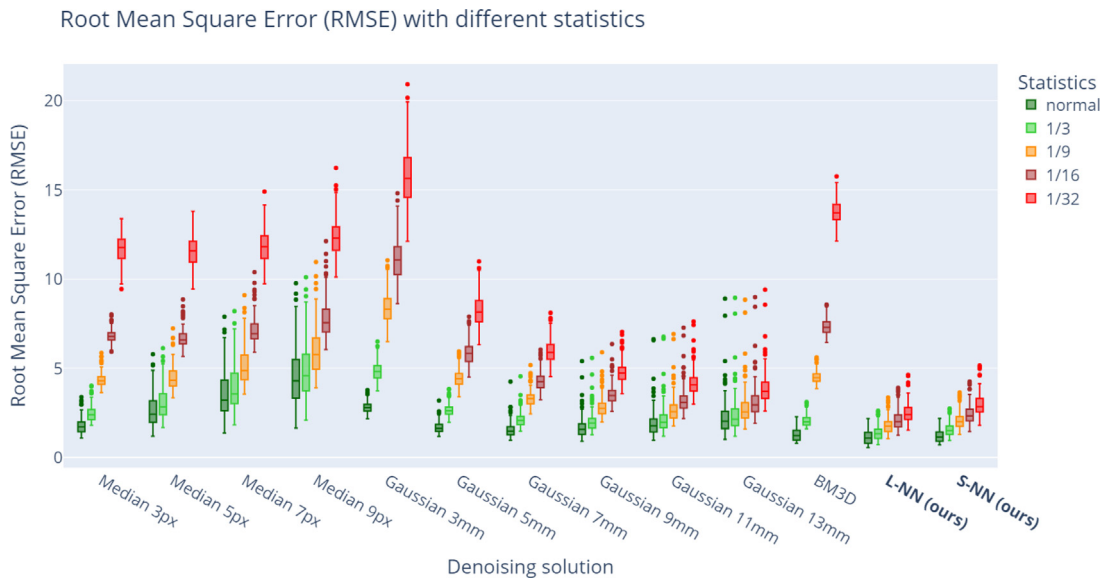
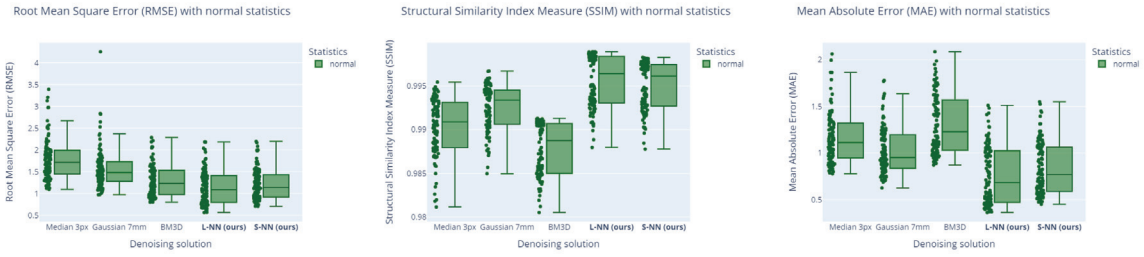
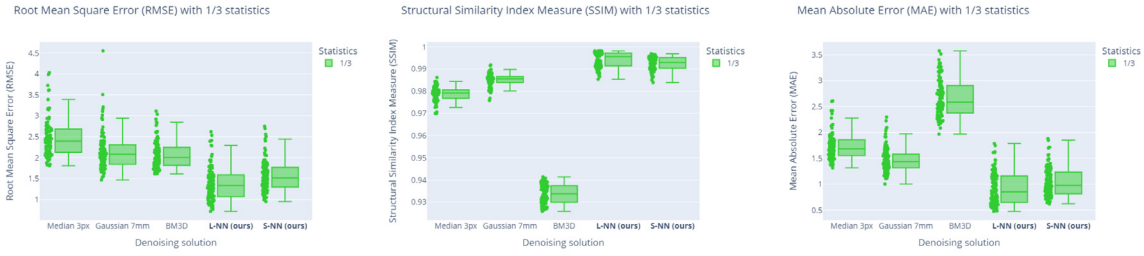



**Fig. 7.** Performance of different filters for RMSE. The graph shows that the performance of the neural network, in addition to having the best values, also has a low standard deviation. Even for images with very low statistics, the scores of the different measures are close, unlike for example the median filter. For detailed description of the box plots see [30,31].




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

**Fig. 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

**Fig. 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.

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 h of accumulation time. Once the planar image was acquired, the filtered image was obtained within 1–2 min 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 Ref. [10], 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 do not 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. Therefore we have been working on the complex clinical evaluation of

**Table 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.

| | | 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 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.

| | | 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 |

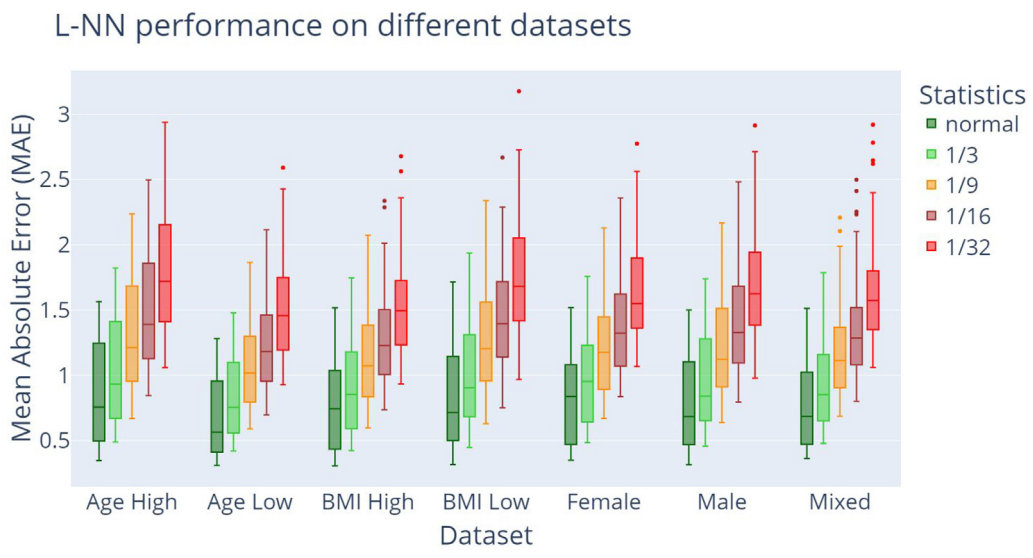



**Fig. 10.** The performance of the L-NN filter computing MAE on different validation sets. The box-plot type graph shows the measured performance for each data set. It can be seen that the median values are nearly the same for all data sets, with a small number of outliers. For detailed description of the box plots see [30,31].

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.

## 6. 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, localization preview scans. The evaluation method presented here can be applied and generalized in all cases where noise-free measurements are not available.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References


[1] M. Elhoseny, K. Shankar, Optimal bilateral filter and convolutional neural network based denoising method of medical image measurements, Meas. J. Int. Meas. Confed. 143 (2019) 125–135, http://dx.doi.org/10.1016/j.measurement.2019.04.072.

[2] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, T. Aila, Noise2Noise: LEarning image restoration without clean data, in: 35th International Conference on Machine Learning, ICML 2018, vol. 7, International Machine Learning Society (IMLS), 2018, pp. 4620–4631, arXiv:1803.04189.

[3] J.r. Jan, Medical Image Processing, Reconstruction and Restoration: Concepts and Methods, CRC Press, 2005, pp. 1–730, http://dx.doi.org/10.1201/9781420030679, URL https://www.taylorfrancis.com/books/mono/10.1201/9781420030679/medical-image-processing-reconstruction-restoration-jiri-jan-jiri-jan.

[4] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising with block-matching and 3D filtering, in: E.R. Dougherty, J.T. Astola, K.O. Egiazarian, N.M. Nasrabadi, S.A. Rizvi (Eds.), Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, vol. 6064, SPIE, 2006, 606414, http://dx.doi.org/10.1117/12.643267, URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.643267.

[5] L. Fan, F. Zhang, H. Fan, C. Zhang, Brief review of image denoising techniques, Vis. Comput. Ind. Biomed. Art 2 (1) (2019) 1–12, http://dx.doi.org/10.1186/s42492-019-0016-7, URL https://vciba.springeropen.com/articles/10.1186/s42492-019-0016-7.

[6] M. Mäkitalo, 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, 2009, pp. 26–32, http://dx.doi.org/10.1109/LNLA.2009.5278406.

[7] A. Foi, Image and video denoising by sparse 3D transformdomain collaborative filtering, Transf. Spectral Methods Group (2014) URL http://www.cs.tut.fi/~foi/GCF-BM3D/index.html#ref_software.

[8] M. Mäkitalo, A. Foi, Optimal inversion of the generalized anscombe transformation for Poisson-Gaussian noise, IEEE Trans. Image Process. 22 (1) (2013) 91–103, http://dx.doi.org/10.1109/TIP.2012.2202675.

[9] P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, Association for Computing Machinery (ACM), 2008, pp. 1096–1103, http://dx.doi.org/10.1145/1390156.1390294.

[10] A. Kovács, G. Légrádi, A. Wirth, F. Nagy, A. Forgács, S. Barna, I. Garai, T. Bükki, A mesterséges és emberi intelligencia értéke a csontszcintigráfia példáján keresztül, Magyar Onkológia 64 (2) (2020) 153–158.

[11] N. Yuan, J. Zhou, J. Qi, Half2Half: deep neural network based CT image denoising without independent reference data, Phys. Med. Biol. 65 (21) (2020) 215020, http://dx.doi.org/10.1088/1361-6560/aba939, URL https://iopscience.iop.org/article/10.1088/1361-6560/aba939.

[12] I.A. Elbakri, J.A. Fessier, Segmentation-free statistical image reconstruction for polyenergetic X-ray computed tomography, in: Proceedings - International Symposium on Biomedical Imaging, 2002-Janua, 2002, pp. 828–831, http://dx.doi.org/10.1109/ISBI.2002.1029387.

[13] G. Last, M. Penrose, Lectures on the Poisson Process, (August) 2017, http://dx.doi.org/10.1017/9781316104477.

[14] C.C. Aggarwal, Neural Networks and Deep Learning, Springer International Publishing, 2018, http://dx.doi.org/10.1007/978-3-319-94463-0.

[15] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408, http://dx.doi.org/10.5555/1756006.1953039.

[16] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90, http://dx.doi.org/10.1145/3065386, URL http://code.google.com/p/cuda-convnet/.

[17] O. Ronneberger, P. Fischer, 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), 9351, Springer Verlag, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28, URL http://lmb.informatik.uni-freiburg.de/http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net.

[18] 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, I. Garai, Q-bot: automatic DICOM metadata monitoring for the next level of quality management in nuclear medicine, EJNMMI Phys. 8 (1) (2021) 1–13, http://dx.doi.org/10.1186/s40658-021-00371-w, URL https://ejnmmiphys.springeropen.com/articles/10.1186/s40658-021-00371-w.

[19] J. Teuwen, N. Moriakov, Convolutional neural networks, in: Handbook of Medical Image Computing and Computer Assisted Intervention, Academic Press, 2019, pp. 481–501, http://dx.doi.org/10.1016/B978-0-12-816176-0.00025-9.

[20] F. Chollet, et al., Keras, 2015, https://keras.io.

[21] 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, X. Zheng, TensorFlow: A system for large-scale machine learning, in: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, USENIX Association, 2016, pp. 265–283, URL arXiv:1605.08695, https://arxiv.org/abs/1605.08695v2.

[22] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, International Conference on Learning Representations, ICLR, 2014, URL arXiv:1412.6980, https://arxiv.org/abs/1412.6980v9.

[23] Y.A. LeCun, L. Bottou, G.B. Orr, K.-R. Müller, Efficient BackProp, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7700 LECTURE NO, Springer, Berlin, Heidelberg, 2012, pp. 9–48, http://dx.doi.org/10.1007/978-3-642-35289-8_3, URL https://link.springer.com/chapter/10.1007/978-3-642-35289-8_3.

[24] A. Horé, D. Ziou, Image quality metrics: PSNR vs. SSIM, in: Proceedings - International Conference on Pattern Recognition, 2010, pp. 2366–2369, http://dx.doi.org/10.1109/ICPR.2010.579.

[25] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612, http://dx.doi.org/10.1109/TIP.2003.819861.

[26] S. van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, T. Yu, the scikit-image contributors, Scikit-image: image processing in python, PeerJ 2 (2014) e453, http://dx.doi.org/10.7717/peerj.453.

[27] Y. Hou, C. Zhao, D. Yang, Y. Cheng, Comments on image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Process. 20 (1) (2011) 268–270, http://dx.doi.org/10.1109/TIP.2010.2052281.

[28] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095, http://dx.doi.org/10.1109/TIP.2007.901238.

[29] D. Minarik, O. Enqvist, E. Trägårdh, Denoising of scintillation camera images using a deep convolutional neural network: A Monte Carlo simulation approach, J. Nucl. Med. 61 (2) (2020) 298–303, http://dx.doi.org/10.2967/jnumed.119.226613, URL http://jnm.snmjournals.org/content/61/2/298.full.

[30] R. McGill, J.W. Tukey, W.A. Larsen, Variations of box plots, Amer. Statist. 32 (1) (1978) 12–16.

[31] Plotly Technologies Inc., Box plots in Python, URL https://plotly.com/python/box-plots/.