JMI Journal of Medical Imaging

RESEARCH PAPER

Denoising digital breast tomosynthesis projections using deep learning with synthetic data as training set

Darlan M. N. de Araújo,* Denis H. P. Salvadeo, and Davi D. de Paula
São Paulo State University, Institute of Geociences and Exact Sciences, Rio Claro, Brazil

ABSTRACT. **Purpose:** Image denoising based on deep neural networks (DNN) needs a big dataset containing digital breast tomosynthesis (DBT) projections acquired in different radiation doses to be trained, which is impracticable. Therefore, we propose extensively investigating the use of synthetic data generated by software for training DNNs to denoise DBT real data.

Approach: The approach consists of generating a synthetic dataset representative of the DBT sample space by software, containing noisy and original images. Synthetic data were generated in two different ways: (a) virtual DBT projections generated by OpenVCT and (b) noisy images synthesized from photography regarding noise models used in DBT (e.g., Poisson–Gaussian noise). Then, DNN-based denoising techniques were trained using a synthetic dataset and tested for denoising physical DBT data. Results were evaluated in quantitative (PSNR and SSIM measures) and qualitative (visual analysis) terms. Furthermore, a dimensionality reduction technique (t-SNE) was used for visualization of sample spaces of synthetic and real datasets.

Results: The experiments showed that training DNN models with synthetic data could denoise DBT real data, achieving competitive results to traditional methods in quantitative terms but showing a better balance between noise filtering and detail preservation in a visual analysis. T-SNE enables us to visualize if synthetic and real noises are in the same sample space.

Conclusion: We propose a solution for the lack of suitable training data to train DNN models for denoising DBT projections, showing that we just need the synthesized noise to be in the same sample space as the target image.

 [DOI: 10.1117/1.JMI.10.3.034001]

Keywords: deep learning; image denoising; digital breast tomosynthesis; synthetic data; virtual clinical trials

Paper 22188GRR received Aug. 2, 2022; revised Mar. 13, 2023; accepted May 1, 2023; published May 22, 2023.

1 Introduction

Digital breast tomosynthesis (DBT)[1] has arisen as a new radiation technique that fills the gaps in digital mammography (DM). While DM has problems with dense breasts and tissue overlapping, DBT aims to acquire several breast projections and reconstruct a three-dimensional (3D) model composed of several slices with reduced tissue overlapping. In this way, the specialist can visualize microcalcifications that DM could hide, increasing the accuracy of detecting lesions. However, like other radiation-based imaging approaches, DBT must use a radiation dose as low

*Address all correspondence to Darlan M. N. de Araújo, darlan.nakamura@unesp.br

2329-4302/2023/$28.00 © 2023 SPIE

as possible, to reduce the risks to the patients. Reducing the radiation dose results in a higher presence of noise in the projections, implying a lower quality of the reconstructed DBT image. Thus, it is necessary to use computational denoising methods to improve the image quality, either applying them in the projection domain (pre-reconstruction step) or directly in the reconstructed DBT images (post-reconstruction step).

Multiple works have studied denoising techniques to improve DBT image quality,[2–4] others have investigated techniques to synthesize DBT virtual images[5] and still others for generating low radiation dose projections from full radiation dose projections,[6] enabling studies without exposing patients to additional scans. With the advent of deep neural networks (DNN), several DNN-based denoising approaches were proposed, showing important results when applied to filter DBT projections.[7]

Considering only the denoising approaches, the DNN-based methods need a training step to learn a model that will be applied to filter noisy DBT data, which requires intensive computation and a big dataset of original and noisy data. On the other hand, the traditional non-DNN approaches can be applied directly to the desired DBT noisy data, without a training step.

In our previous work found in Ref. 7, it was shown that DNN-based blind denoising methods trained and tested on synthetic data achieved results similar or better than the best traditional denoising methods, even with a limited dataset size and computational power. These synthetic image datasets were generated in two different ways: (a) by a software simulating DBT virtual clinical trials and (b) by synthesizing noisy versions of the general images from a photography dataset. The reason to evaluate some DNN-based denoising methods in the second case (b) was to verify how these methods perform on noise models commonly used for modeling DBT projections but not evaluated yet for them in the literature [e.g., denoising CNN (DnCNN)[8] was not evaluated for denoising images corrupted by Poisson–Gaussian noise].

Thus, due to the relevant results of DNN approaches achieved by Ref. 7 and from other works in the literature, and the lack of real data available for training DNN methods, this paper has the following objetives: (1) evaluate if trained DNN-based denoising methods in synthetic datasets generated by software (virtual dataset or photography dataset) can achieve suitable results when tested in real DBT image datasets (physical dataset); (2) compare the combination of several cost-functions for DNN-based denoising methods; and (3) evaluate the sample space of noisy data of the different datasets (synthetic, virtual, and physical) using a data visualization technique.

The main contributions of this paper can be summarized as follows:

- DNN-based blind denoising methods trained on synthetic data simulating noise models of DBT were suitable to denoise real DBT projections. In fact, the models trained on the virtual phantoms dataset and those models trained on the photography dataset with synthetic noise had similar results, showing that DNN-based denoising methods do not depend on the image content itself, but sample spaces of both synthetized and real noises must be similar, enabling other medical imaging systems to be trained on general images;
- Data visualization technique proved to be a useful tool to explore the sample space of noisy data of the different datasets, which is important to clarify the similarity (distribution) of training and test datasets;
- Cost functions defined by a combination of complementary loss functions can improve the results of DBT projection denoising, achieving a better balance in both quantitative and qualitative terms, especially in microcalcification areas.

The remainder of this work is organized as follows: Sec. 2 presents a brief literature review related to our proposal. Section 3 describes the datasets used in the experiments and the methodology. Section 4 presents the results achieved in the experiments. Finally, Sec. 5 discusses the experimental results and presents the conclusions and proposals for future works.

2 Related Works

In the literature, noise in DBT projections has been modeled as the following distributions: Poisson,[2] Gaussian,[3] or Poisson–Gaussian.[4]

Several works have been studying how to apply denoising methods to improve DBT projection quality. To bring a brief review of these methods, this section was organized in traditional (non-DNN-based) methods (Sec. 2.1) and DNN-based methods (Sec. 2.2). In addition, other related works of DNN-based methods for denoising but applied for other domains or imaging modalities are discussed in Sec. 2.3.

2.1 Traditional Methods for Denoising DBT Projections

Reference 2 performed the noise filtering in the projections using Gaussian denoising algorithms combined with the Anscombe transform,[9] concluding that the transform improved the quality of the image. Reference 3 analyzed the viability of two non-local algorithms to denoise DBT projections, finding that the method block matching and 3D collaborative filtering (BM3D)[10] is more effective than the non-local means (NLM).[11]

Reference 4 proposed a pipeline with five steps to apply any denoising technique to filter additive white Gaussian noise in the projections: (1) remove the calibration, (2) variance stabilization by using the generalized Anscombe transform to convert from Poisson–Gaussian noise to a signal-independent Gaussian noise, (3) apply a denoising method suitable to Gaussian noise, (4) apply the inverse of the generalized Anscombe transform to return to the original domain, and (5) insert the calibration. After the last step, the filtered image is achieved. The authors showed that the calibration step combined with the generalized Anscombe transform improved the result of Gaussian denoising methods.

An extensive comparison of non-DNN denoising methods in DBT projections was made by Ref. 12. For the noise considered as Poisson (i.e., the Anscombe transform was not applied), the denoising methods used were: pointwise maximum *a posteriori* (MAP)[13] and Poisson non-local means (P-NLM).[14] For the noise considered as Gaussian (i.e., Anscombe transform used), the methods were non-local means (NLM), BM3D, pointwise Wiener filter (PWF),[15] generalized Wiener filter (GWF),[16] isotropic Wiener filter (IWF),[17] and separable Wiener filter (SWF).[17] The best results were achieved by BM3D. The authors also compared the denoising applied in different steps: (a) pre-reconstruction step (i.e., on projection domain), which is the focus of this paper; (b) post-reconstruction step (i.e., on reconstructed image domain); (c) double filtering (i.e., denoising in both pre- and post-reconstruction steps). The best result was achieved when the noise filtering was performed only on projections.

2.2 DNN-Based Methods for Denoising DBT Projections

Recently, DNNs have been applied to denoise DBT projections, especially due to the limitation that the traditional methods are attached to the noise model (i.e., non-blind filtering) and cannot be applied when the noise is unknown.

The work of Liu et al.[18] proposed a DNN-based method called neural network convolution (NNC) to improve DBT projections in terms of noise and artifact reduction. They used a cadaver phantom in which projections were acquired by a DBT system (Selenia Dimensions, Hologic, Inc., Bedford, Massachusetts). For training the NNC, they used the projections acquired with 25% of protocol dose as noisy data and projections with 200% of protocol dose as ground truth, expecting that the NNC learn how to map low-quality to high-quality projections. They evaluated the NNC in terms of structural similarity index measure (SSIM),[19] achieving the best results in comparison with bilateral filtering, BM3D, and K-singular value decomposition (K-SVD).

Reference 20 proposed a DNN-based method trained with adversarial loss to denoise DBT projections generated by OpenVCT,[5] which is a software that generates DBT virtual clinical trials. They used the generated virtual phantoms for training and physical phantoms for testing and compared the proposed method with K-SVD[21] and BM3D using the metrics peak signal-to-noise ratio (PSNR)[19] and HaarPSI. Although the proposed denoising approach reached the best result using the HaarPSI metric, it achieved third place among the compared methods using the PSNR metric. In general, the results show that simulated data (generated by virtual clinical trials software) used as a training dataset has high potential, but lacks a greater quantity of comparison methods.

Two previous works of ours investigated the use of DNN-based denoising methods. Reference 22 focused on denoising DBT projections using convolutional neural networks (CNNs) by training and testing on virtual DBT data generated by OpenVCT. In turn, the work

of Ref. 7 extends the previous paper by presenting a benchmark of DNN-based denoising approaches when the training and test datasets are composed of small patches from virtual DBT projections. Both works showed that DNN-based methods achieved the best results in comparison to traditional methods.

For summarizing and establishing the state-of-the-art methods for denoising DBT projections found in the literature, Tables 1 and 2 show the PSNR and SSIM[19] of methods using virtual phantoms and physical phantoms, respectively. These measures are widely used to evaluate denoising approaches and are better explained in Sec. 3.4. The prefix AT+ means the Anscombe

Table 1 Comparison of denoising methods to filter projections of virtual phantoms. Bold PSNR values indicate the highest values in each column.

Method	Reference	PSNR	SSIM
AT + adaptive Wiener filter	2, 23	**39.65**	**0.931**
Adaptive Wiener filter	2, 23	35.65	0.828
DnCNN	7	33.93	0.86
WST	7	33.41	0.79
DnCNN10	7	31.75	0.88
CNN	7	30.46	0.78
AT+PWF	12	29.91	0.51
AT+SWF	12	29.80	0.50
AT+IWF	12	29.79	0.50
AT+GWF	12	29.64	0.51
BM3D	7	29.49	0.79
NLM	7	29.07	0.76
AT+BM3D	12	28.68	0.40
MAP	12	27.89	0.41
Autoencoder	7	27.72	0.78
AT+NLM	12	26.94	0.27
DnCNN	22	26.92	0.70
P-NLM	12	26.83	0.26
CGAN	7	26.19	0.71
BM3D	22	25.93	0.62
NLM	22	24.98	0.50
WST	22	24.96	0.56
K-SVD	7	23.56	0.77
Wiener filter	7	21.84	0.75
Wiener filter	22	21.70	0.57
DIP	7	21.0	0.80
K-SVD	20	20.16	—
GAN	20	18.91	—
BM3D	20	18.44	—
MLP	7	7.31	0.09

Table 2 Comparison of denoising methods to filter pre-reconstruction images (on the projection domain) of physical phantoms. The references in the last four lines did not calculate PSNR and SSIM values. Bold PSNR values indicate the highest values in each column.

Method	Reference	PSNR	SSIM
AT+BM3D	12	**30.58**	0.36
AT+PWF	12	30.11	0.40
AT+IWF	12	30.08	0.39
AT+SWF	12	30.03	0.39
AT+GWF	12	29.54	0.40
MAP	12	28.83	0.36
AT+NLM	12	28.41	0.26
P-NLM	12	28.25	0.24
NNC	18	—	**0.97**
K-SVD	18	—	0.94
BM3D	18	—	0.93
Bilateral filter	18	—	0.91
NLM	3	—	—
BM3D	3	—	—
Calibration+AT+BM3D	4	—	—
AT+RF3D	6	—	—

transform was applied before the denoising method and the inverse of the Anscombe tranform was applied after it. Note that the dataset and methodology used in each paper (row) are different, meaning that this is the reason for the PSNR and SSIM obtained by some papers being higher than others.

Observing the references in Table 1, the works of Refs. 2 and 23 used 18 virtual phantoms, in which each phantom has 15 projections generated by the system GE DBT (Senographe DS, General Electric Healthcare, Chalfront St. Giles, United Kingdom), totaling 270 projections. The work of Ref. 12 used the software OpenVCT[5] to generate 15 virtual phantoms with 15 projections each, totaling 225 projections. Reference 20 used 30 virtual phantoms with 25 projections each, generated by the OpenVCT, totaling 750 projections. References 22 and 7 also used OpenVCT to generate 100 exams with 15 projections each. There is an evident difficulty in directly comparing each work due to the difference in the methodology and datasets. Nevertheless, we can note the high performance of the Wiener filter and its variations.

Observing Table 2, Refs. 3, 6, and 12 used an anthropomorphic phantom prototyped by CIRS, Inc. (Reston, Virginia), and the acquisition of projections was made by the DBT Selenia Dimensions scanner, Hologic, Bedford, Massachusetts. Reference 18 used a cadaver phantom and a DBT Selenia Dimensions scanner, Hologic, Bedford, Massachusetts, to acquire the projections. To the test, they used 10 clinic cases at the DBT system of the University of Iowa. In Table 2, we can note the BM3D applied after the Anscombe transform achieved the best results in terms of PSNR, followed by the variations of Wiener filter on the Anscombe domain. In addition, the highest SSIM of 0.97 was achieved by the method NNC, which is a DNN-based approach.

2.3 Other Related DNN-Based Works for Denoising

Other works also evaluated the training of DNN-based denoising methods using simulated data, but not for DBT projections. For instance, Ref. 24 proposed a training procedure using two datasets for denoising DBT reconstructed images: (i) the first composed by an anthropomorphic

breast model generated by VICTRE project[25] and virtually scanned by an x-ray imaging simulation tool, developed by GE Global Research, named Computer Assisted Tomography SIMulator2 (CatSim);[26] (ii) a physical phantom made with tissue-mimicking materials to simulate a real breast, imaged by a real DBT imaging system. The use of synthetic datasets made it possible to obtain low dose (LD) images and high dose (HD) images. In the training, the DNN received as input the LD images and expected as output a denoised image similar to the corresponding HD image. For training, they used a combination of mean squared error (MSE) and adversarial loss functions between the DNN output and the corresponding HD image. All the training was made using patches of the reconstructed breast image. The trained DNN model was tested using real human subject DBT containing biopsy-proven invasive ductal carcinomas (masses) and ductal carcinomas *in situ* (microcalcification clusters). The results showed that the proposed training process using virtual and physical phantoms made the denoiser applicable to denoise the real human subject DBT, with competitive results.

In Ref. 27, the authors of Ref. 24 used a DNN-based denoising method to evaluate the influence of the target image quality in the DNN training. The datasets used were the same of Ref. 24: (i) the training set was the digital breast phantoms and (ii) the test set was the physical phantom. This work found competitive results in this cross domain evaluation and also found that the DNN-based denoiser is heavely dependent of target image quality.

Other instance was Ref. 28 that used a dataset with real DBT data[29] and artificial noise to train a denoising model, in which presented good results at restoring the DBT reconstructed images.

Finally, works that studied the training and evaluation of DNN-based denoising with synthetically noise addition for other imaging modalities, e.g., magnetic reasonance. This is the case of Ref. 30 that used a dataset composed of prostate mp-MRI real image data, acquired by two different scanners. They proposed a DNN to improve the quality of MRI images. The model was trained using the images with Gaussian and Rician noise added, whereas the test data were the original data without noise. The denoising method achieved competitive results.

3 Materials and Methods

This section presents the approach of this work and the main resources used. In Sec. 3.1, the data used in the experiments are described. In Sec. 3.2, the denoising methods used in the experiments are explained. In Sec. 3.3, the tools, programming languages, and frameworks used to generate the results are detailed. Finally, in Sec. 3.4, the methodology and experimental setup are detailed.

3.1 Data

The projection datasets (Sec. 3.1.1) are the source of our experiments. To train the DNNs, we processed the projections (images), splitting them into patches (Sec. 3.1.2). Furthermore, Sec. 3.1.3 presents the regions with microcalcifications used in the experiments for a qualitative evaluation.

3.1.1 *Image datasets*

This work used three image sources to perform the experiments, which are described below:

- Berkeley segmentation dataset (BSD):[31] contains 500 gray-scale images of a large variety, such as natural images and images containing people, animals, and objects. This dataset is out of the medical image domain, and it is not related to DBT projections, but in Sec. 4, the reader will observe relevant results using these images. Artificial noise (such as Gaussian, Poisson, and Poisson–Gaussian distribution) was synthesized on the original images to generate noisy versions of them. The parameters used for the Gaussian noise were mean $\mu = 0$ and standard deviation $\sigma = 0.1$. On the Poisson–Gaussian noise, the Poisson noise was applied before the Gaussian noise. Therefore, there are the noisy versions (original image with simulated noise) and the ground truth (original image). Table 3 shows the projection databases. Note that in the row related to BSD (second row) the columns "realization" and "aquisition" are empty because it contains only general images.

Table 3 Number of realizations and acquisition type in each dataset.

Name	Realizations	Acquisition type
BSD	—	—
Alvarado	100	Virtual
Benson	1	Virtual
Burke	1	Virtual
Burnett	1	Virtual
Drake	1	Virtual
Hurst	1	Virtual
Steele	1	Virtual
100%	5	Real
85%	5	Real
50%	5	Real

- Virtual DBT Dataset: Virtual DBT projections were generated using the software OpenVCT,[5] which provided the generation of realistic images simulating a variety of radiation dose, noise, breast tissue density, breast lesions, number of projections, spacing between projections, etc. Furthermore, this tool allows the reconstruction of the projections resulting in the 3D virtual breast. In this study, OpenVCT was used to create the virtual breasts and to simulate the DBT acquisition process (exam) of these breasts by a virtual DBT scanner, generating versions of each projection with and without noise.

As done by Ref. 7, this work used seven virtual DBT datasets: (i) Alvarado, (ii) Benson, (iii) Burke, (iv) Burnett, (v) Drake, (vi) Hurst, and (vii) Steele. All these virtual datasets follow the Selenia Dimensions (Hologic Inc., Bedford, Massachusetts) architecture, acquiring 15 projections in an arc of 15-deg, following an equally spaced angular trajectory above the breast. Each projection from each exam is a gray-scale image with 1792×2048 pixels as dimensions. Table 3 shows these seven datasets (in lines 3–9). Alvarado is the only one that contains 100 realizations, meaning that there are 100 different virtual breast phantoms, each of them containing 15 projections with noisy and ground truth versions. The other virtual DBT datasets contain only one virtual breast phantom, with 15 noisy and noise-free projections.

- Physical DBT Dataset: We used physical artificial phantoms with specific materials to simulate the human breasts generated by Ref. 3. Unlike the virtual DBT dataset, where the samples and the acquisition are completely generated using software, here the object that mimics the breast is scanned by a real DBT equipment. This approach allows us to carry out experiments using the noise generated by a real DBT equipment. However, because just one physical phantom was used, the low variety of the internal breast structure is a limitation. In this dataset, the phantom was scanned 16 times with a protocol dose by Selenia Dimensions (Hologic Inc., Bedford, Massachusetts) equipment. The dimension of each projection is 2048×1664 px.

To perform the experiments, the phantom was also scanned using lower doses (85% and 50%) from the protocol radiation dose (100%), commonly used in real exams, resulting in three dataset variations. The phantom was scanned five times for 85% and five times for 50% from the protocol radiation dose. Table 4 presents the radiation exposure (in uAs) and the entry dose (in mGy) for the scanning. Figure 1 presents a patch from the three first projections from each variation (50%, 85%, and 100% dosis) and the ground truth. Lines 9 to 12 of Table 3 show four datasets. Note that 85% and 50% have five realizations, which means there are five different scans from the same object. The ground truth is the result of the mean of 11 of the 16 realizations acquired with protocol radiation dose (100%), representing the noiseless realization of the object.

Table 4 Radiation exposition and the entry dose for the physical DBT dataset.

Name	Exposition (uAs)	Dose (mGy)
100%	60,000	5.62
85%	51,000	4.79
50%	30,000	2.81

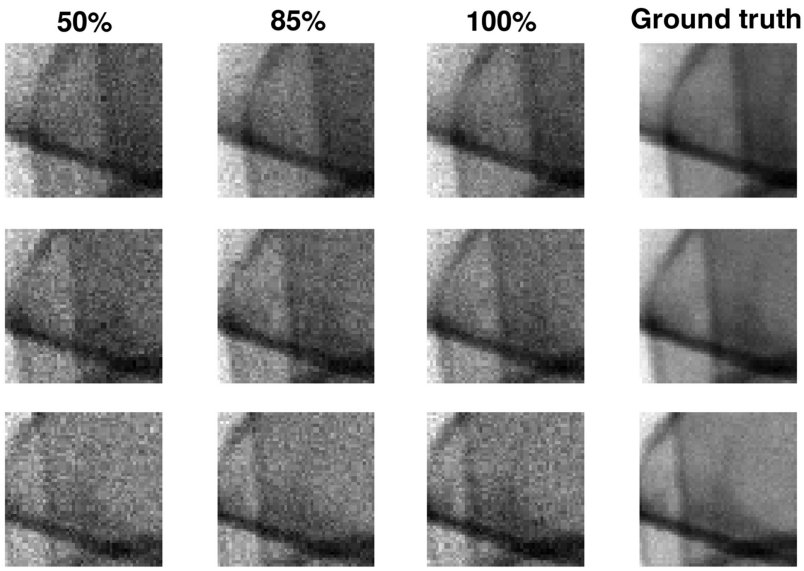

Fig. 1 A patch from the first three projections from each physical DBT dataset variation and ground truth. Each path has 52×52 px.

This will be used as the target in the test phase for all experiments described in the remaining of this work. The five remaining realizations were used as samples (input for DNN) in some experiments in the training phase. Note that the images used to construct the ground truth and the images used as training samples in the 100% dataset were different. The images used to construct the ground truth were never used as a training set.

The target patches of physical dataset used to calculate and backpropagate the error back to the DNN were extracted from the ground truth in both training and test phases. However, in the training phase, the patches were extracted from a different region than in the test phase. Thus, even though the targets in the training and test phases come from the same ground truth, they are different images for the DNN.

3.1.2 *Patch datasets*

As seen in the previous section, the projections are large images and would require a huge computational capability to train DNN models with a large sample size. Therefore, 52×52 px patches were extracted for each image. For images in the virtual and physical DBT projection datasets, we extracted patches from the central breast region. From a single DBT projection, we extracted 200 patches, creating the patch datasets shown in Table 5, where we trained all the DNN models. The column "image dataset" indicates the dataset used to create the corresponding patch dataset.

The virtual patch dataset (row 5 in Table 5) is an aggregation of all seven virtual DBT datasets shown in Table 3, including Alvarado, where the patches were shuffled and selected randomly. This division was made to analyze if only Alvarado would be able to improve the results, because it has 100 virtual breast models, whereas the other ones have only one model.

Table 5 Patch datasets used to train the deep learning models, with the number of patches and the projection dataset where the patches were extracted.

Patch dataset	Training set size	Image dataset
Gaussian	20,000	BSD (with Gaussian noise)
Poisson	20,000	BSD (with Poisson noise)
Poisson–Gaussian	20,000	BSD (with Poisson–Gaussian noise)
Alvarado	20,000	Virtual DBT dataset
Virtual	19,200	Virtual DBT dataset
Mixed-dataset	19,200	Physical and virtual DBT datasets
Mixed-physical	19,200	Physical DBT dataset
50%	19,200	Physical DBT (50% from protocol radiation dose)

To explore the influence of mixed datasets in the experiments, mixed-physical means the partitions 50%, 80%, and 100% of the protocol radiation dose of the physical DBT dataset were mixed in the same proportion (33.33%), shuffled, and randomly selected. Finally, mixed-dataset was built as a mix between the physical DBT dataset and the virtual DBT dataset, in which sample patches were collected randomly.

3.1.3 *Regions of interest*

We extracted regions with microcalcifications from the physical DBT projection dataset since these regions have lower contrast in noisier projections. From the 15 projections, we extracted two regions of interest of 300×300 pixels, as shown in Fig. 2, showing the region of interest (ROI) for the first projection of an exam. In this figure, the red ROI will be referenced as ROI 1 and the yellow as ROI 2.

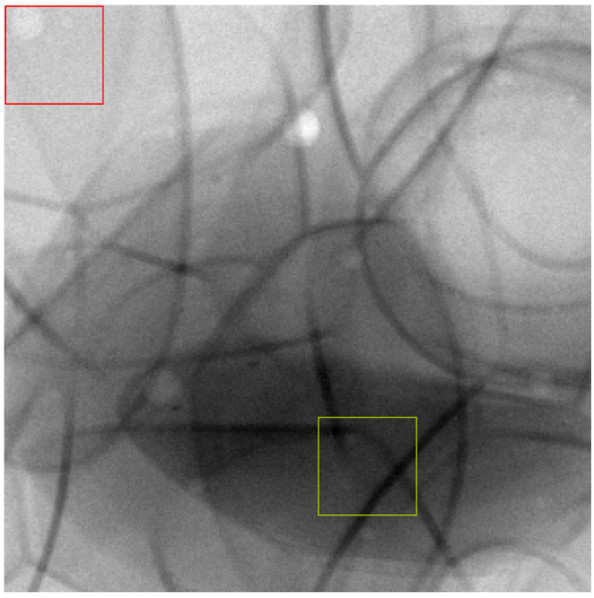

Fig. 2 Regions of interest from a breast region with some microcalcifications. This breast region was extracted from the first projection of one example with 85% of the protocol radiation dose in the physical DBT dataset.

3.2 Methods

In this work, several methods for DBT denoising were used for comparison, as specified in Secs. 3.2.1 and 3.2.2 for non-trainable and trainable denoising methods, respectively.

3.2.1 *Non-trainable denoising methods*

These denoising methods are already available to be applied directly to filter the noisy image, with the drawback that they are not blind methods (except DIP), meaning that they need extra parameters to estimate the noise. The methods used in this work are BM3D,[10] NLM,[32] K-SVD,[21] wavelet soft-thresholding (WST),[33] PWF,[15] and deep image prior (DIP).[34] Although the DIP is based on a neural network, it does not need to be trained, which is why it is placed in this category.

These methods were chosen due to the reasons explained as follows. BM3D was chosen because it is compared as a benchmark in almost all the related works. NLM and Wiener are justified because of the experiments of Ref. 12. K-SVD is an efficient sparse method used as a benchmark on Ref. 20 and WST is a simpler sparse method.

About the implementation details, K-SVD was executed with $k = 11$, patch size $= 5 \times 5$, and 15 iterations. NLM was executed with patch size $= 7 \times 7$, search area size $= 11 \times 11$, and $h = 0.1$. DIP was tested using 100 iterations.

3.2.2 *Trainable denoising methods*

These denoising methods need to be trained in a dataset before being applied to filter the noisy image, with the advantage of blind methods. The methods used in this work are denoising CNN (DnCNN)[8] with 19 layers, DnCNN with 10 layers (DnCNN10), CNN with five layers, CGAN,[35] autoencoder,[36] and multilayer perceptron with five layers (MLP).[37,38]

These trainable methods were chosen due to the reasons explained as follows. DnCNN was applied to denoise DBT projections on Refs. 7 and 22 achieving good results. Autoencoder was applied because it usually reaches good results in denoising task. GAN is used because Ref. 20 proposes a GAN for denoising DBT projections. Since we do not have the source code of the proposed method in Ref. 20, a general GAN was used with sigmoid cross-entropy as cost function. Likewise, we used CNN because Ref. 18 proposes to use a CNN to denoise DBT projections. The MLP was also compared since its the simpler feedforward neural network.

About implementation details, they were trained in the 52×52 px patches from the patch datasets (as described in Sec. 3.1.2). In addition, the images in both training and test phases are normalized linearly to interval [0, 1] and the implementation hyperparameters are described in Table 6. A validation set was not used in any experiment shown in this paper, and training was stopped at epoch 40 without early stopping technique.

3.3 Tools

All experiments were developed mostly in the programming language Python, using the machine learning libraries Numpy,[39] SciPy,[40] Scikit Image,[41] TensorFlow,[42] PyTorch,[43] Keras, and others.

Table 6 Hyperparameters of the neural networks applied in this work.

Network model	Layers	Learning rate	Optimizer	Cost function	Batch size
MLP	5	0.001	Adam	MSE	128
CNN	5	0.001	Adam	MSE	128
DnCNN	19	0.0001	Adam	MSE	128
DnCNN10	10	0.0001	Adam	MSE	128
Autoencoder	10	0.001	Adam	MSE	128
CGAN	14	0.0001	Adam	SCE	256

The source code is freely available to the research community (Code available in a Github repository: https://github.com/darlannakamura/masters-degree).

3.4 Methodology and Experimental Setup

The experiments were divided into four parts: (1) comparing different training datasets (Sec. 3.4.1), (2) comparing cost functions for training DNN-based denoising models (Sec. 3.4.2), (3) dataset visualization (Sec. 3.4.3), and (4) visualization of ROIs (Sec. 3.4.4). All the results are analyzed in terms of SSIM[19] and PSNR[44] for quantitative results, and also in a visual way for qualitative results. SSIM is used to evaluate the structural difference between two images (in our case the denoised image and the ground truth) and is related to the human visual perception. The latter is an objective metric frequently used to evaluate denoising algorithms. The methodology used in each experiment is described below.

3.4.1 *Comparing different training datasets*

The goal of the experiments whose results are in Sec. 4.1 is to compare the influence of the training datasets, including datasets of different domains (e.g., BSD, which is not related to medical images and virtual DBT dataset), assessing the results of reducing noise in the physical DBT dataset. The models trained in the patch datasets named Gaussian, Poisson, Poisson–Gaussian, and Alvarado are the same models trained by Ref. 7.

To test the trainable methods, the patches of 52×52 px were submitted to each denoising method and the result patches were aggregated again, resulting in a 300×300 px image. It is important to mention that the patches for the testing stage were extracted from the larger image considering some overlapping among them. In fact, they are necessary because when we reconstruct the patches to create the 300×300 px image, we need to overlap them together by an aggregation algorithm. The aggregation algorithm used here is simple: we just apply the mean of pixels in the place when two patches are overlapped. Note that the DBT tomographic reconstruction was not performed, i.e., all experiments were realized in the projection domain. Each denoising method was evaluated in terms of PSNR and SSIM, calculated only in the 300×300 px aggregated final image. Figure 3 shows this process.

In turn, to test the non-trainable methods, the filtering process was applied directly in the 300×300 px region, as shown in Fig. 4. Note that the test region was the 300×300 px region instead of the individual 52×52 px patches, because the non-local methods (i.e., BM3D and

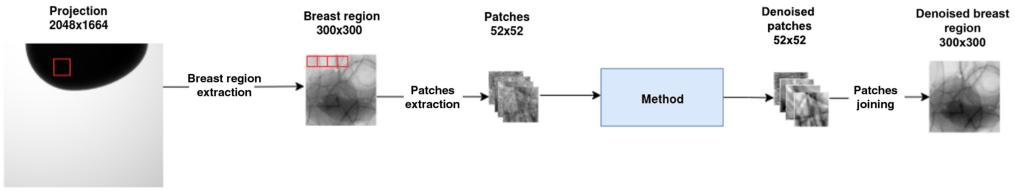

Fig. 3 Test pipeline of trainable DNN-based denoising methods.

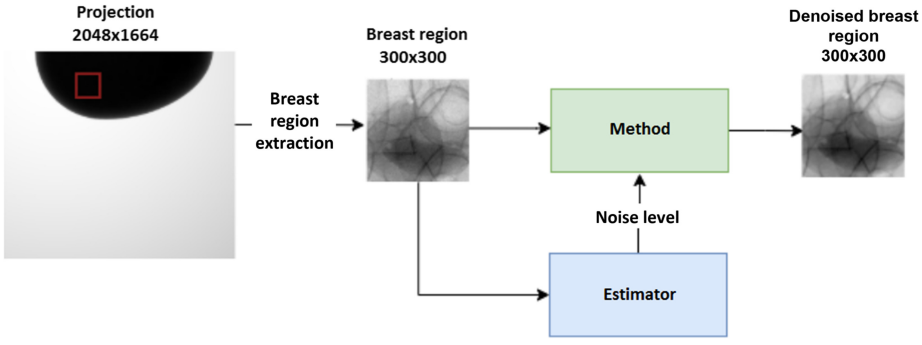

Fig. 4 Test pipeline of non-trainable denoising methods.

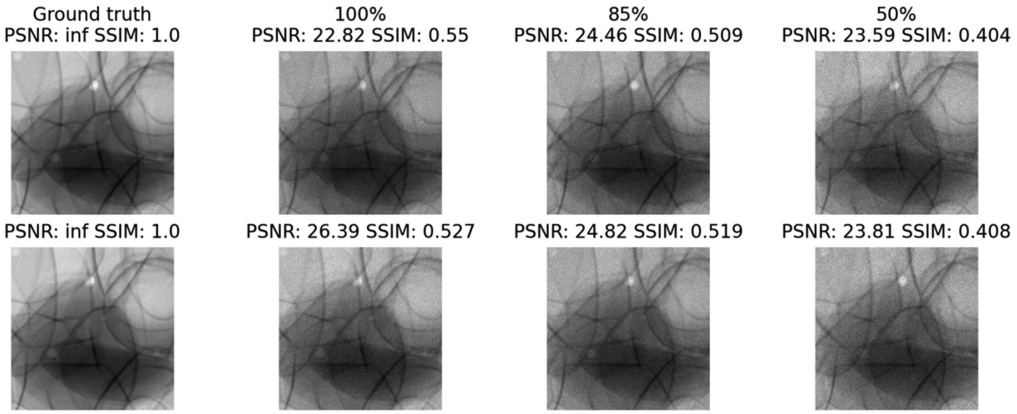

Fig. 5 Some instances of patches from physical DBT dataset. The first column is the ground truth and the other columns show the images acquired with different radiation doses (100%, 85%, and 50% from the protocol dose radiation) in the physical DBT dataset.

NLM) may be impaired due to the small area available since they naturally use the whole image for the filtering process. Note also that in the test process of non-trainable methods (Fig. 4), the noise standard deviation is estimated by the method proposed by Ref. 45. Both categories (trainable and non-trainable methods) were tested on the physical DBT dataset (Sec. 3.1.1) using the 300×300 px region.

Figure 5 shows 2 of the 15 regions used as tests from the physical DBT dataset. Each region was extracted from one projection. As the physical DBT datasets were acquired from one physical phantom, varying the radiation dose, it is possible to have paired results as shown in Fig. 5. The first row shows the first region and the second row the second region. Column 1 shows the ground truth; columns 2 to 4 show versions of each patch acquired with 100%, 85%, and 50% of the protocol radiation dose. In the ground truth column, the SSIM value is 1.0 and the PSNR is *inf* because this is the ideal image. In the other columns, where the images contain noise, comparing the PSNR and SSIM with the ground truth, we can note that the images in the last column have more noise and inferior results for PSNR and SSIM values.

3.4.2 *Cost functions comparison*

In turn, the goal of the experiments whose results are in Sec. 4.2 aims to compare different cost functions in the training phase of the denoising methods DnCNN, DnCNN10, CNN, and Autoencoder. The cost functions used were: (1) mean absolute error (MAE),[46] 2) MSE,[19] (3) SSIM, and (4) mutiscale SSIM (MS-SSIM).[47] More details of these cost functions can be found in Appendix A.

The combination of these cost functions was also tested by simply adding them to create another one. The combinations, represented by the symbol +, are: (1) MAE+MSE, (2) SSIM+MAE, (3) SSIM+MSE, (4) MS-SSIM+SSIM, (5) MS-SSIM+MAE, and (6) MS-SSIM+MSE. The training was performed using 500,000 patches from the Alvarado dataset. This test aims to verify if there exists one or a combination of two cost functions that can improve the DBT projection denoising.

3.4.3 *Exploration of datasets using data visualization technique*

The goal of this experiment is to visualize the sample spaces of noise from the training set of datasets Alvarado, physical dataset (50%, 85%, 100% of the protocol radiation dose), Gaussian, Poisson, and Poisson–Gaussian through the dimensionality reduction method t-distributed stochastic neighbor embedding (t-SNE),[48] which enable us to compare the noise distribution in a spatial manner.

For each dataset, patches were selected and the noise data were obtained by subtracting the ground truth patch from the noisy patch, defining d-dimensional points in sample space of noise,

where d is the patch size. The t-SNE was executed using two components, perplexity 40 and 800 iterations to reduce the point dimensions from d to only 2, enabling a visualization in a two-dimensional (2-D) cartesian plane. The results of this experiment are found in Sec. 4.3.

3.4.4 *Analysis of regions of interest*

The last experiment aims to apply the tested denoising methods in some special patches, focusing on assessing visually the methods in microcalcification regions, whose selected patches and denoising results are in Sec. 4.4. This section will present the RoI with breast lesions, and the capability of methods to preserve the details.

4 Results

The results section is organized as follows. Section 4.1 will discuss the comparison results of the denoising methods: trainable or non-trainable methods. Section 4.2 will present the results with different cost functions. Section 4.3 will describe an exploration using a data visualization technique. Section 4.4 will analyze the results qualitatively.

4.1 Comparing Different Training Datasets

Considering the trainable denoising methods, it was selected just three models to present their results here: DnCNN, DnCNN10 and CNN as shown by Ref. 7, which were the methods capable of producing satisfactory results. These models were trained by Ref. 7, meaning that each dataset represents the training set used to train the model.

To summarize these results, Fig. 6 shows the mean value of PSNR and SSIM reached by DnCNN, DnCNN10, and CNN for each training dataset. Note that, in this Fig. 6, the methods were tested in the 50% of the protocol radiation dose dataset from Physical DBT Dataset (Sec. 3.1.1). The samples in the training phase are different from the test phase. More details and results can be seen in Appendix B, which contains all results for each (trainable and non-trainable) denoising methods evaluated in this work,

Observing Fig. 6, the methods trained in Poisson, Gaussian, Poisson-Gaussian, and Alvarado datasets have superior results compared to the others. In terms of SSIM, these four datasets achieved a mean of 0.714, whereas the other datasets are below 0.586. In terms of PSNR, the first datasets achieved mean values above 27.56 dB, and the other achieved mean values below 23.87 dB.

The models trained in synthetic BSD and virtual DBT datasets reached better results than the models trained on the dataset with DBT physical exams. In other words, we found that models trained on the dataset different from the medical image domain with synthetic noise were able to reach better quantitative results than the models trained on the same dataset domain of the target domain (in this case physical DBT dataset) as the test set.

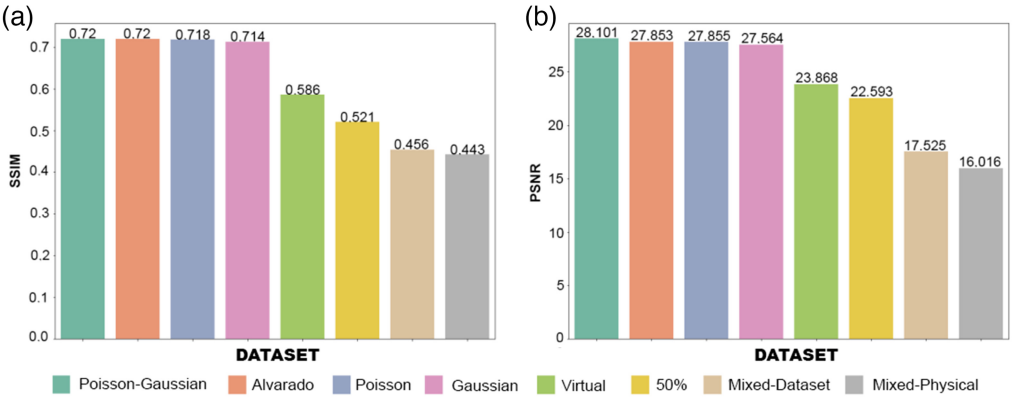

Fig. 6 Mean of (a) SSIM and (b) PSNR for the models DnCNN, DnCNN10, and CNN, trained in different datasets.

Table 7 Mean and standard deviation of SSIM and PSNR for the DnCNN, DnCNN10, CNN, and autoencoder trained on the samples of 100% of the protocolar radiation dose in physical dataset. The values are ordered in descending order by SSIM.

Ranking	Cost function	SSIM	PSNR (dB)
1	SSIM+MSE	0.745 ± 0.06	25.95 ± 3.73
2	MSSSIM+MAE	0.741 ± 0.07	27.43 ± 3.94
3	SSIM+MAE	0.740 ± 0.06	26.59 ± 3.54
4	MSSSIM+SSIM	0.739 ± 0.06	23.66 ± 3.90
5	MSSSIM+MSE	0.737 ± 0.08	27.66 ± 4.81
6	MSSSIM	0.733 ± 0.07	24.56 ± 7.22
7	MAE	0.733 ± 0.07	26.41 ± 3.46
8	MSE	0.732 ± 0.07	25.49 ± 2.86
9	MAE+MSE	0.700 ± 0.07	22.28 ± 6.21
10	SSIM	0.681 ± 0.11	20.57 ± 8.12

4.2 Cost Functions Comparison

This section summarizes the results of using different cost functions used during the training step of some trainable denoising methods. Table 7 shows the mean and standard deviation of PSNR and SSIM of the methods DnCNN, DnCNN10, CNN, and Autoencoder, trained on the samples of 100% of the protocol radiation dose in Physical DBT Dataset and tested in the ground truth.

Observing the Table 7, the cost function that achieved the best results in terms of SSIM is SSIM+MSE (0.745). However, the cost function MSSSIM+MSE presented the best PSNR value (27.66 dB). On the other hand, the MSSSIM+MAE cost function presented a better balance between PSNR and SSIM, with an SSIM difference of just 0.004 from the best cost function in terms of SSIM and a PSNR difference of just 0.23 dB from the best cost function in terms of PSNR.

In turn, Table 8 shows the mean and standard deviation of PSNR and SSIM of the methods DnCNN, DnCNN10, CNN, and Autoencoder trained on the samples of 85% of the protocol radiation dose in Physical DBT Dataset and tested in the ground truth.

Table 8 SSIM and PSNR mean and standard deviation for the DnCNN, DnCNN10, CNN, and autoencoder trained on the samples of 85% of the protocol radiation dose in physical DBT dataset. The values are ordered in descending order by SSIM.

Ranking	Cost function	SSIM	PSNR (dB)
1	SSIM+MSE	0.742 ± 0.05	26.18 ± 3.90
2	MSSSIM+MAE	0.739 ± 0.06	27.58 ± 3.74
3	SSIM+MAE	0.738 ± 0.06	26.81 ± 3.46
4	MSSSIM+SSIM	0.737 ± 0.06	23.96 ± 4.16
5	MSSSIM+MSE	0.734 ± 0.07	27.36 ± 4.28
6	MAE	0.731 ± 0.07	26.70 ± 3.42
7	MSE	0.731 ± 0.07	25.86 ± 2.90
8	MSSSIM	0.731 ± 0.07	24.86 ± 7.19
9	MAE+MSE	0.705 ± 0.07	22.82 ± 5.88
10	SSIM	0.680 ± 0.10	20.54 ± 7.91

Table 9 SSIM and PSNR mean and standard deviation for the DnCNN, DnCNN10, CNN, and autoencoder trained on the samples of 50% of the protocol radiation dose in physical DBT dataset. The values are ordered in descending order by SSIM.

Ranking	Cost function	SSIM	PSNR (dB)
1	SSIM+MSE	0.724 ± 0.04	26.23 ± 3.50
2	MSSSIM+MAE	0.722 ± 0.05	27.68 ± 3.65
3	SSIM+MAE	0.721 ± 0.05	26.87 ± 3.27
4	MSSSIM+SSIM	0.720 ± 0.05	24.05 ± 4.13
5	MSE	0.717 ± 0.06	26.31 ± 3.14
6	MSSSIM+MSE	0.717 ± 0.06	27.17 ± 3.91
7	MSSSIM	0.714 ± 0.06	24.97 ± 6.58
8	MAE	0.713 ± 0.05	26.51 ± 3.11
9	MAE+MSE	0.709 ± 0.06	24.98 ± 3.28
10	SSIM	0.669 ± 0.09	20.56 ± 7.62

Analyzing the Table 8, as before, the cost function SSIM+MSE achieved the best result in terms of SSIM (0.742), whereas the cost function MSSSIM+MAE presented the best PSNR value (27.58 dB), being the one that achieved the better balance between both metrics, with an SSIM difference of just 0.003 from the SSIM+MSE.

Regarding noisier DBT data, Table 9 shows the mean and standard deviation of PSNR and SSIM of the methods DnCNN, DnCNN10, CNN, and Autoencoder trained on the samples of 50% of the protocol radiation dose in Physical DBT Dataset and tested in the ground truth.

Analyzing the Table 9, the cost function SSIM+MSE achieved the best value in terms of SSIM (0.724), similar to previous cases. The cost function MSSSIM+MAE presented the best value in terms of PSNR (27.68 dB) and the better balance in terms of both measures, staying just 0.002 of SSIM behind the SSIM+MSE.

Independently of the dataset (100%, 85% or 50% of the protocol radiation dose), the same three cost functions achieved the best results in terms of SSIM and PSNR were SSIM+MSE, MSSSIM+MAE, and SSIM+MAE. In this way, the MSSSIM+MAE is the cost function with a better balance between both metrics, reaching the highest PSNR and a close SSIM to the highest one.

Observe that in the Tables 7, 8, and 9 when the cost function MSE is used alone, poor results are generated. This happens because the MSE takes into account only the difference between pixels, ignoring the structural differences between images. On the other hand, when the cost function SSIM is used alone, we can observe that the poorest results are generated. This happens because SSIM will consider only the luminance, contrast, and structure, ignoring the pixel-to-pixel relation, which makes the model insensitive to big changes in the pixel intensity. Alone, these two cost functions are poorly ranked in the former tables, but when used together, the MSE calculates the quadratic difference between pixels and therefore solves the pixel-to-pixel problem that occurs in SSIM. Thus, as can be seen in the former tables, the cost function SSIM+MSE achieved the best values in terms of SSIM. A similar procedure occurs with the MS-SSIM.

In Tables 11, 12 and 13 in Appendix B, there are the complete results comparing all methods with all cost functions used, regarding the partitions of the Physical DBT Dataset with 100%, 85% and 50% of the protocol radiation dose, respectively.

4.3 Exploration of Dataset using Data Visualization Technique

According to the results from Sec. 4.1, where the models trained on synthetic and virtual DBT datasets reached the best results, we proposed a visual exploration to understand the noise from these datasets. In this way, consider the noise points extracted as explained in Sec. 3.4.3. Then,

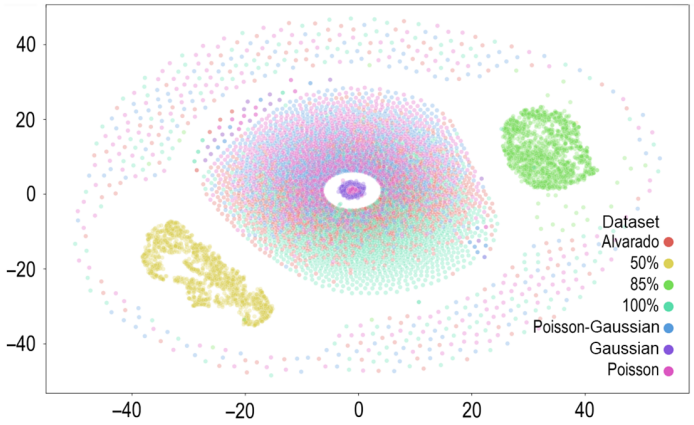

Fig. 7 Noise visualization of the datasets using t-SNE.

with the noise points of all patches of different datasets, we applied the t-SNE method (also described in Sec. 3.4.3) to generate the visualization shown in Fig. 7. So, each point in the Figure represents the extracted 2-D noise point of each patch of a specific dataset.

The datasets shown in Fig. 7 were selected because they achieved the best result in experiments from Sec. 4.1. Analyzing Fig. 7, it is possible to note that the points for partitions containing 50% and 85% (yellow and green points, respectively) of the protocol radiation dose in the physical DBT dataset are far from each other, whereas the points for Poisson (pink points), Gaussian (blue points), and the 100% (light-green points) of the protocol radiation dose in physical DBT dataset are partially mixed. This can explain why the models presented poor results when trained in the 50% of the protocol radiation dose in physical DBT dataset in Sec. 4.1 (Fig. 6), because the sample space of the training dataset is very different from the test dataset. Furthermore, the datasets Poisson-Gaussian, Poisson, and Alvarado are very mixed with the 100% (target sample space) of the protocol radiation dose dataset and may explain the good results in Fig. 6.

4.4 Analysis of Regions of Interest

This section describes and discusses qualitatively the denoising methods' performance over an ROI extracted from the examples in the physical DBT dataset. These regions were manually selected since they are low-contrast regions with microcalcifications, one kind of lesion looked for by radiologists in breast cancer diagnosis, being a good test to evaluate the denoising algorithms under real conditions.

Please note that the purpose of this work is to improve the noise of real medical images using images with simulated noise as a training set. The focus is not on enhancing lesion detectability. However, in the examples presented in this section, we use ROIs with lesions to compare the noise in the original image with the denoised image. It should be noted that no conclusions can be made regarding the improvement of the lesion detectability rate using denoising methods. This may be a topic for future research using a dedicated experimental configuration tailored for this aim, in which images or patches are exhibited to a medical specialist who is unaware of the denoising approach. The specialist would be directed to identify any anomalies, and the identification performance could be assessed and compared among the different methods. Another approach that could be employed is to utilize a pretrained method for lesion detection to identify lesions in both the noisy and denoised images, and subsequently evaluate if there is any improvement in detection performance when using the denoised images.

Figure 8 shows the denoising results for the methods DnCNN, DnCNN10, CNN, Autoencoder, K-SVD, BM3D, PWF, CNN, CGAN, NLM, and DIP applied over the ROI 1, using the same region of the same projection in one example with 85% of the protocol radiation dose in physical DBT dataset. The denoising methods DnCNN, DnCNN10, CNN, and autoencoder were trained using the best cost functions in terms of SSIM. In Fig. 8, it can be perceived that the trainable methods DnCNN and DnCNN10 better preserved the breast structure and the smaller microcalcification at the right, whereas K-SVD and PWF keep too much noise. The CNN

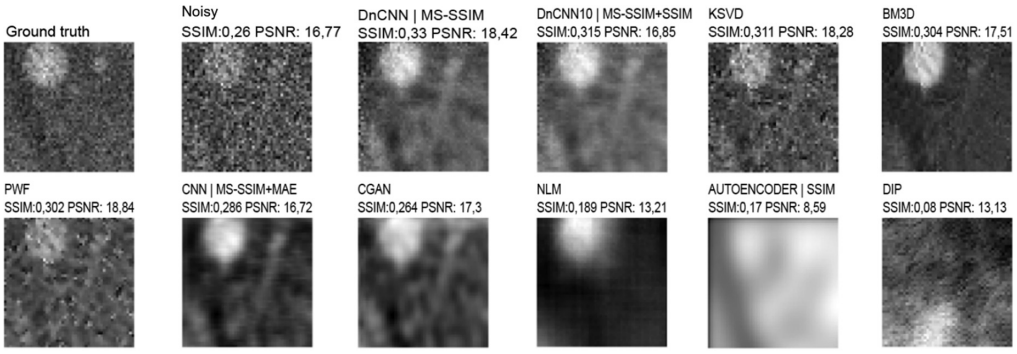

Fig. 8 The results from ROI 1 in one example with 85% of the protocol radiation dose in physical DBT dataset.

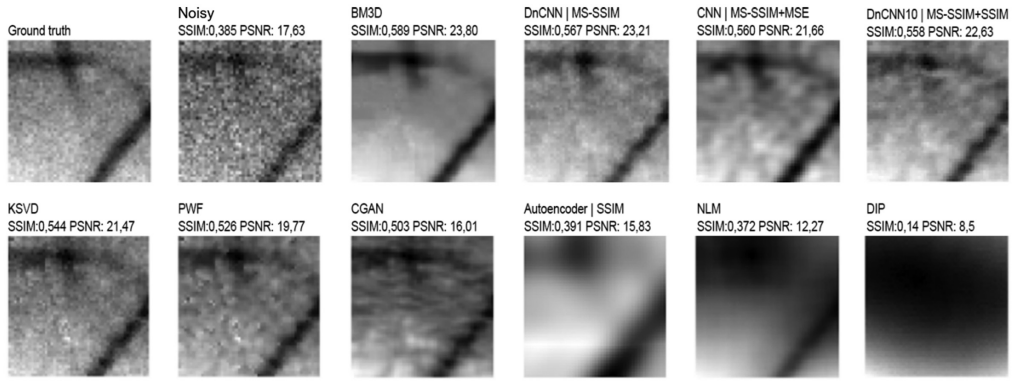

Fig. 9 The results from ROI 2 in one example with 85% of the protocol radiation dose in physical DBT dataset.

and CGAN showed an image with better contrast, whereas the methods NLM, autoencoder, and DIP do not provide good images. In turn, Fig. 9 shows the ROI 2 over the same projection in one example with 85% of the protocol radiation dose in the physical DBT dataset. Using the denoising method BM3D, the microcalcification is not clear, but using the methods CNN, DnCNN, and DnCNN10 it is possible to distinguish it. Using the methods CGAN, autoencoder, NLM, and DIP, the resulting image was not good and using the methods K-SVD and PWF the resulting image remained noisy.

Figures 10 and 11 show the results for the same two ROIs over the same region but now on the example with 50% of the protocol radiation dose in physical DBT dataset, in which has more noise. In these figures, no methods were able to maintain the contrast in the smaller microcalcification at the right to be sufficiently visible in Fig. 11. This shows that if a real

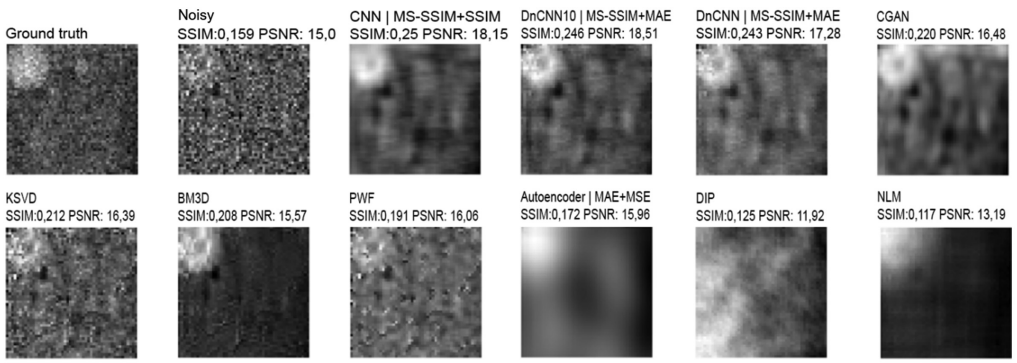

Fig. 10 Results for ROI 1 at one example with 50% of the protocol radiation dose in physical DBT dataset.

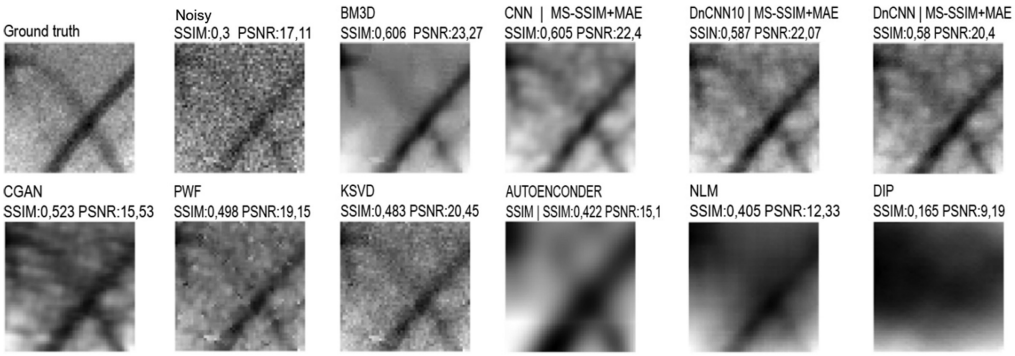

Fig. 11 Results for ROI 2 at one example with 50% of the protocolar radiation dose in physical DBT dataset.

Table 10 Quantitative results of each denoising method over ROIs, using the metrics SSIM and PSNR (ordered descending by SSIM). Results consider examples with 100% and 85% of the protocol radiation dose in the physical DBT dataset. Bold PSNR values indicate the highest values in each column.

Ranking	Method	SSIM	PSNR
1	BM3D	**0.439**	**19.90**
2	KSVD	0.406	19.12
3	DnCNN	0.401	18.10
4	CNN	0.396	17.78
5	PWF	0.392	18.36
6	DnCNN10	0.392	18.41
7	CGAN	0.387	16.63
8	Entry	0.294	16.66
9	NLM	0.270	12.90
10	Autoencoder	0.252	12.27
11	DIP	0.115	10.34

exam was made using these denoising methods, using half of the protocol radiation dose, it may exist some microcalcification that the professional can not observe. Therefore, this case is not recommended.

Thus, taking into account the performance of DnCNN and DnCNN10 in the qualitative analysis, these denoising methods displayed better results than the non-trainable methods for preserving breast structure and enhancing the microcalcifications, especially compared with BM3D in the ROIs presented before, even performing better quantitatively, as one can see in Table 10 in row 2. This table shows the mean results of SSIM and PSNR taking into account only the ROI 1 and ROI 2 in examples with 100% and 85% of the protocol radiation dose in physical DBT dataset, where the methods BM3D and K-SVD achieved the best results. The first achieved a mean value of SSIM of 0.038 and PSNR of 1.80 dB greater than DnCNN.

5 Discussion and Conclusions

In this work, we evaluated five traditional methods and six DNN-based methods for denoising DBT projections. Through the results, we concluded that neural networks trained on synthetic data could beat the others trained in real physical DBT data.

As well, the results from the models trained on synthetic datasets and virtual DBT datasets were similar, highlighting the capability of blind denoising neural networks to learn the noise distribution from the training data.

By visualizing the sample spaces of noise points of training and target datasets, we can conclude that the essential condition to perform well using a synthetic data as training dataset for denoising DBT projections is that the training dataset must be generated by using a noise model from the same sample space of the target noisy image. This finding is especially important because the medical images lack real data acquired in different conditions to be able to train DNN-based denoising models because it is inappropriate to perform multiple exams with different radiation doses from real patients, due to excessive radiation exposure.

Regarding the comparison between traditional methods and DNN-based methods, BM3D achieved the highest values of SSIM in test data, whereas DnCNN with a combination of cost functions performed better qualitatively, keeping the breast structure and improving the contrast in microcalcification regions, making the microcalcification more visible, reducing the chance of misleading and consequently a bad diagnosis (false negatives).

Finally, future works include: (a) a study of the distribution of the noise extracted in each epoch of training in trainable methods; (b) a simulation of different noise levels in OpenVCT data by improving the noise model embedded in this software; (c) investigation of noise spaces in synthetic training and test sets varying the noise level; (d) evaluation of DnCNN method using synthetic data for training in other contexts, noise distributions, and imaging systems (e.g., remote sensing data, ultrasound, magnetic resonance); (e) depth investigation if the DNN proposed methods would generate artifacts that can affect the diagnosis.

6 Appendix A

This section presents the equations of the cost functions used in the experiments. Consider the following variables for these cost funcions: the ground truth image y, the predicted (filtered) image $\hat{y}$ obtained by a trainable DNN-based denoising technique, and $M \times N$ image size.

6.1 Absolute Squared Error (MAE)

$$\mathrm{MAE}(y,\hat{y}) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|y_{i,j} - \hat{y}_{i,j})|. \tag{1}$$

6.2 Mean Squared Error (MAE)

$$\mathrm{MSE}(y,\hat{y}) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(y_{i,j} - \hat{y}_{i,j})^2. \tag{2}$$

6.3 SSIM

$$C^{\mathrm{SSIM}}(y,\hat{y}) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(1 - \mathrm{SSIM}(y_{i,j},\hat{y}_{i,j})). \tag{3}$$

The SSIM(.) in the equation is the Structural Similarity Index Measure.[19] This entire equation is used only when the DNN is trained.

6.4 Multi-Scale SSIM (MS-SSIM)

$$\text{MS-SSIM}(y,\hat{y}) = 1 - \text{MS-SSIM}(y,\hat{y}), \tag{4}$$

being MS-SSIM

$$\text{MS-SSIM}(y,\hat{y}) = [l(y_M,\hat{y}_M)].\prod_{i=1}^{M}[c(y_i,\hat{y}_i)][s(y_i,\hat{y}_i)], \tag{5}$$

where $[l(y_M,\hat{y}_M)]$ is a sub-sampling operator, $[c(y_i,\hat{y}_i)]$ is the contrast operator, and $[s(y_i,\hat{y}_i)]$ is the structure operator. More details can be found in Ref. 47.

7 Appendix B

This section shows all individual quantitative results obtained for each denoising method evaluated in this work.

In Tables 11–13, we can found the individual denoising results for the first experiment (comparision of trainable and non-trainable denoising methods) tested on physical DBT dataset with 100%, 85%, and 50% of the protocol radiation dose, respectively, which was summarized for each training dataset in Sec. 4.1.

In turn, Table 14 shows individual denoising results for the second experiment (comparison of cost functions of DnCNN) evaluated on the physical DBT dataset, which was summarized for each cost function in Sec. 4.2.

Table 11 Results from DNN-based denoising methods training in different datasets and tested on the physical DBT dataset with 100% of the protocol radiation dose using the 300×300 px region (on projection domain). Bold PSNR values indicate the highest values in each column.

Position	Method	Training dataset	SSIM	PSNR (dB)
1	DnCNN10	Alvarado	**0.772**	**30.84**
2	CNN	Poisson–Gaussian	0.768	28.33
3	CNN	Gaussian	0.766	27.60
4	CNN	Poisson	0.766	27.90
5	DnCNN10	Poisson–Gaussian	0.754	29.35
6	DnCNN10	Poisson	0.753	28.94
7	DnCNN10	Gaussian	0.749	28.56
8	CGAN	Virtual	0.734	25.30
9	DnCNN	Poisson–Gaussian	0.730	29.77
10	DnCNN	Poisson	0.728	29.38
11	DnCNN	Alvarado	0.725	29.81
12	CGAN	Mixed-dataset	0.724	21.31
13	DnCNN	Gaussian	0.722	29.13
14	CNN	Alvarado	0.709	24.07
15	CGAN	Gaussian	0.707	18.16
16	CGAN	Poisson–Gaussian	0.705	18.13
17	CGAN	50%	0.688	21.51
18	CGAN	Poisson	0.671	17.33
19	DnCNN10	Virtual	0.657	25.84
20	CNN	Virtual	0.635	22.79
21	Autoencoder	Poisson–Gaussian	0.631	19.87
22	Autoencoder	Gaussian	0.63	19.03
23	Autoencoder	Poisson	0.629	18.73
24	Autoencoder	Alvarado	0.627	22.57
25	Autoencoder	Virtual	0.624	22.85
26	CNN	50%	0.571	20.43
27	DnCNN10	50%	0.565	25.55
28	DnCNN	Virtual	0.557	24.58
29	CNN	Mixed-dataset	0.548	18.04
30	CNN	Mixed-physical	0.518	14.13

Table 11 (*Continued*).

Position	Method	Training dataset	SSIM	PSNR (dB)
31	DnCNN	50%	0.516	24.04
32	Autoencoder	50%	0.503	16.44
33	Autoencoder	Mixed-dataset	0.486	11.48
34	Autoencoder	Mixed-physical	0.472	10.59
35	DnCNN	Mixed-physical	0.467	16.98
36	DnCNN	Mixed-dataset	0.466	17.89
37	DnCNN10	Mixed-dataset	0.414	16.75
38	DnCNN10	Mixed-physical	0.403	17.14
39	MLP	50%	0.346	17.18
40	MLP	Poisson–Gaussian	0.299	15.23
41	MLP	Virtual	0.299	15.56
42	MLP	Gaussian	0.295	13.76
43	MLP	Mixed-dataset	0.212	13.47
44	MLP	Alvarado	0.195	12.58
45	MLP	Mixed-physical	0.187	13.07
46	CGAN	Mixed-physical	0.151	10.45
47	MLP	Poisson	0.101	8.19
48	CGAN	Alvarado	0.078	9.42

Table 12 Results from DNN-based denoising methods training in different datasets and tested on the physical DBT dataset with 85% of the protocol radiation dose using the 300×300 px region (on projection domain). Bold PSNR values indicate the highest values in each column.

Position	Method	Training dataset	SSIM	PSNR (dB)
1	DnCNN10	Alvarado	**0.766**	**30.55**
2	CNN	Poisson–Gaussian	0.763	27.73
3	CNN	Poisson	0.761	27.4
4	CNN	Gaussian	0.761	27.15
5	DnCNN10	Poisson–Gaussian	0.74	28.12
6	DnCNN10	Poisson	0.736	27.74
7	DnCNN10	Gaussian	0.732	27.34
8	CGAN	Mixed-dataset	0.718	21.36
9	CNN	Alvarado	0.71	24.37
10	DnCNN	Poisson–Gaussian	0.709	28.66
11	DnCNN	Alvarado	0.708	29.27
12	DnCNN	Poisson	0.707	28.77
13	DnCNN	Gaussian	0.701	28.39
14	CGAN	Poisson–Gaussian	0.695	18.04
15	CGAN	Gaussian	0.693	18.04
16	CGAN	50%	0.68	21.47
17	CGAN	Poisson	0.658	17.19

Table 12 (*Continued*).

Position	Method	Training dataset	SSIM	PSNR (dB)
18	CNN	Virtual	0.637	22.98
19	Autoencoder	Poisson–Gaussian	0.632	20.11
20	Autoencoder	Gaussian	0.632	19.22
21	DnCNN10	Virtual	0.631	25.53
22	Autoencoder	Poisson	0.631	18.9
23	Autoencoder	Alvarado	0.628	22.54
24	Autoencoder	Virtual	0.626	22.98
25	CNN	50%	0.577	20.86
26	CNN	Mixed-dataset	0.548	17.91
27	DnCNN10	50%	0.543	24.61
28	DnCNN	Virtual	0.528	24.12
29	CNN	Mixed-physical	0.516	14.0
30	Autoencoder	50%	0.512	17.12
31	DnCNN	50%	0.491	23.32
32	Autoencoder	Mixed-dataset	0.486	11.39
33	Autoencoder	Mixed-physical	0.472	10.54
34	DnCNN	Mixed-physical	0.45	16.95
35	DnCNN	Mixed-dataset	0.446	17.69
36	DnCNN10	Mixed-dataset	0.395	16.7
37	DnCNN10	Mixed-physical	0.386	17.04
38	MLP	50%	0.295	15.65
39	MLP	Virtual	0.283	15.41
40	MLP	Poisson–Gaussian	0.281	14.95
41	MLP	Gaussian	0.277	13.54
42	MLP	Mixed-dataset	0.198	13.34
43	MLP	Alvarado	0.182	12.48
44	MLP	Mixed-physical	0.172	12.95
45	MLP	Poisson	0.095	8.09

Table 13 Results from DNN-based denoising methods training in different datasets and tested on the physical DBT dataset with 50% of the protocol radiation dose using the 300×300 px region (on projection domain). Bold PSNR values indicate the highest values in each column.

Position	Method	Training dataset	SSIM	PSNR (dB)
1	DnCNN10	Alvarado	**0.734**	**29.79**
2	CNN	Gaussian	0.727	27.69
3	CNN	Poisson	0.726	27.97
4	CNN	Poisson–Gaussian	0.726	28.23
5	CGAN	Alvarado	0.713	26.12
6	CNN	Alvarado	0.705	24.67
7	CGAN	Mixed-dataset	0.69	21.69

Table 13 (*Continued*).

Position	Method	Training dataset	SSIM	PSNR (dB)
8	CGAN	Mixed-dataset	0.685	26.07
9	CGAN	Mixed-physical	0.674	21.53
10	CGAN	Virtual	0.673	25.72
11	DnCNN10	Poisson–Gaussian	0.648	26.71
12	DnCNN10	Poisson	0.639	26.38
13	CNN	Virtual	0.637	23.11
14	CGAN	Gaussian	0.637	19.04
15	Autoencoder	Gaussian	0.635	20.82
16	Autoencoder	Poisson	0.635	20.42
17	Autoencoder	Poisson–Gaussian	0.634	21.66
18	DnCNN10	Gaussian	0.633	26.24
19	DnCNN	Alvarado	0.63	26.83
20	Autoencoder	Alvarado	0.626	22.33
21	Autoencoder	Virtual	0.625	23.24
22	CGAN	Poisson	0.611	18.31
23	DnCNN	Poisson	0.605	26.06
24	DnCNN	Poisson–Gaussian	0.605	26.03
25	DnCNN	Gaussian	0.598	26.0
26	CNN	50%	0.576	20.81
27	CNN	Mixed-dataset	0.547	17.95
28	DnCNN10	Virtual	0.534	23.19
29	CNN	Mixed-physical	0.512	13.88
30	Autoencoder	50%	0.51	17.28
31	Autoencoder	Mixed-dataset	0.488	11.49
32	Autoencoder	Mixed-physical	0.473	10.62
33	DnCNN10	50%	0.449	22.61
34	DnCNN	Virtual	0.426	21.88
35	DnCNN	50%	0.395	21.21
36	DnCNN	Mixed-physical	0.389	17.03
37	DnCNN	Mixed-dataset	0.378	18.18
38	DnCNN10	Mixed-dataset	0.341	16.87
39	DnCNN10	Mixed-physical	0.328	17.09
40	MLP	50%	0.239	14.36
41	MLP	Virtual	0.227	14.2
42	MLP	Poisson–Gaussian	0.219	14.83
43	MLP	Gaussian	0.215	13.71
44	MLP	Mixed-dataset	0.154	12.87
45	MLP	Alvarado	0.141	11.96
46	MLP	Mixed-physical	0.126	12.6
47	MLP	Poisson	0.075	8.02

Table 14 DnCNN trained with different cost functions and compared with the methods: BM3D, K-SVD, PWF, NLM, DIP, and CGAN. Bold PSNR values indicate the highest values in each column.

Dataset	Method	Cost function	SSIM	PSNR (dB)
—	Noisy image	—	0.402	21.6
100	BM3D	—	**0.785**	29.61
100	DnCNN	MS-SSIM	0.781	30.68
100	DnCNN	SSIM+MSE	0.779	28.41
100	DnCNN	SSIM	0.779	29.64
100	DnCNN	MS-SSIM+MAE	0.778	30.33
100	DnCNN	MS-SSIM+SSIM	0.777	28.18
100	DnCNN	SSIM+MAE	0.776	29.2
100	DnCNN	MS-SSIM+MSE	0.776	**30.92**
100	DnCNN	MAE	0.776	29.47
100	DnCNN	MSE	0.768	27.09
100	CGAN	—	0.727	21.57
100	K-SVD	—	0.727	27.67
100	PWF	—	0.72	26.42
100	DnCNN	MAE+MSE	0.666	14.85
100	NLM	—	0.665	23.76
100	DIP	—	0.531	19.52
85	BM3D	—	**0.782**	29.89
85	DnCNN	MS-SSIM	0.776	**30.86**
85	DnCNN	SSIM+MSE	0.775	29.28
85	DnCNN	MS-SSIM+SSIM	0.772	28.24
85	DnCNN	MS-SSIM+MAE	0.772	29.61
85	DnCNN	SSIM	0.771	28.46
85	DnCNN	SSIM+MAE	0.771	29.36
85	DnCNN	MS-SSIM+MSE	0.77	30.03
85	DnCNN	MAE	0.769	29.19
85	DnCNN	MSE	0.765	26.94
85	CGAN	—	0.737	22.58
85	K-SVD	—	0.71	26.97
85	PWF	—	0.704	25.57
85	DnCNN	MAE+MSE	0.683	15.75
85	NLM	—	0.665	23.9
85	DIP	—	0.533	19.62
50	BM3D	—	**0.763**	29.3
50	DnCNN	SSIM+MSE	0.747	28.96
50	DnCNN	MSE	0.747	28.35
50	DnCNN	MS-SSIM+MAE	0.747	**30.14**
50	DnCNN	MS-SSIM	0.746	29.56
50	DnCNN	SSIM+MAE	0.745	29.16
50	DnCNN	MS-SSIM+SSIM	0.744	27.28

Table 14 (*Continued*).

Dataset	Method	Cost function	SSIM	PSNR (dB)
50	DnCNN	MS-SSIM+MSE	0.742	28.95
50	DnCNN	SSIM	0.741	26.43
50	DnCNN	MAE	0.74	28.2
50	DnCNN	MAE+MSE	0.732	23.68
50	CGAN	—	0.719	23.70
50	NLM	—	0.657	23.87
50	PWF	—	0.645	23.97
50	K-SVD	—	0.643	24.3
50	DIP	—	0.533	19.7

Disclosures

The authors declare that they have no conflict of interest.

Acknowledgments

We thank Dr. Andrew Maidment and Dr. Bruno Barufaldi from the University of Pennsylvania for supplying the DBT virtual dataset.

References

1. L. T. Niklason et al., "Digital tomosynthesis in breast imaging," *Radiology* **205**(2), 399–406 (1997).
2. M. A. Vieira, P. R. Bakic, and A. D. Maidment, "Effect of denoising on the quality of reconstructed images in digital breast tomosynthesis," *Proc. SPIE* **8668**, 86680C (2013).
3. M. A. Vieira et al., "Feasibility study of dose reduction in digital breast tomosynthesis using non-local denoising algorithms," *Proc. SPIE* **9412**, 94122C (2015).
4. L. R. Borges et al., "Pipeline for effective denoising of digital mammography and digital breast tomosynthesis," *Proc. SPIE* **10132**, 1013206 (2017).
5. B. Barufaldi et al., "OpenVCT: a GPU-accelerated virtual clinical trial pipeline for mammography and digital breast tomosynthesis," *Proc. SPIE* **10573**, 1057358 (2018).
6. L. R. Borges et al., "Restoration of low-dose digital breast tomosynthesis," *Meas. Sci. Technol.* **29**(6), 064003 (2018).
7. D. M. N. de Araújo, D. H. P. Salvadeo, and D. D. de Paula, "A benchmark of denoising digital breast tomosynthesis in projection domain: neural network-based and traditional methods," *Proc. SPIE* **12032**, 1203207 (2022).
8. K. Zhang et al., "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017).
9. F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," *Biometrika* **35**(3/4), 246–254 (1948).
10. K. Dabov et al., "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007).
11. A. Buades, B. Coll, and J.-M. Morel, "Non-local means denoising," *Image Process. On Line* **1**, 208–212 (2011).
12. D. C. Scarparo et al., "Evaluation of denoising digital breast tomosynthesis data in both projection and image domains and a study of noise model on digital breast tomosynthesis image domain," *J. Med. Imaging* **6**(3), 031410 (2019).
13. E. D. S. Ribeiro, "Novas propostas em filtragem de projeções tomográficas sob rudo Poisson," (2010).
14. C.-A. Deledalle, F. Tupin, and L. Denis, "Poisson NL means: unsupervised non local means for Poisson noise," in *IEEE Int. Conf. Image Process.*, IEEE, pp. 801–804 (2010).
15. D. T. Kuan et al., "Adaptive noise smoothing filter for images with signal-dependent noise," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-7**(2), 165–177 (1985).
16. A. L. Levada and N. D. Mascarenhas, "Filtragem adaptativa de rudo gaussiano em imagens através da minimização da informação de fisher observada," in *VI Workshop de Visão Computacional (WVC)*, pp. 7–12 (2010).

17. S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Inc. (1993).
18. J. Liu et al., "Radiation dose reduction in digital breast tomosynthesis (DBT) by means of deep-learning-based supervised image processing," *Proc. SPIE* **10574**, 105740F (2018).
19. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).
20. P. Sahu et al., "Using virtual digital breast tomosynthesis for de-noising of low-dose projection images," in *IEEE 16th Int. Symp. Biomed. Imaging (ISBI 2019)*, IEEE, pp. 1647–1651 (2019).
21. M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006).
22. D. M. de Araújo, D. H. Salvadeo, and D. D. De Paula, "Denoising digital breast tomosynthesis projections using convolutional neural networks," *Proc. SPIE* **11596**, 115961L (2021).
23. M. A. D. C. Vieira et al., "Investigating Poisson noise filtering in digital breast tomosynthesis," in *IX Workshop de Visão Computacional-WVC, IX*, Universidade Federal Fluminense (UFF), pp. 1–6 (2013).
24. M. Gao, J. A. Fessler, and H.-P. Chan, "Deep convolutional neural network with adversarial training for denoising digital breast tomosynthesis images," *IEEE Trans. Med. Imaging* **40**(7), 1805–1816 (2021).
25. A. Badano et al., "Evaluation of digital breast tomosynthesis as replacement of full-field digital mammography using an *in silico* imaging trial," *JAMA Network Open* **1**, e185474–e185474 (2018).
26. B. D. Man et al., "CatSim: a new computer assisted tomography simulation environment," *Proc. SPIE* **6510**, 65102G (2007).
27. M. Gao, J. A. Fessler, and H.-P. Chan, "Digital breast tomosynthesis denoising using deep convolutional neural network: effects of dose level of training target images," *Proc. SPIE* **11595**, 115951K (2021).
28. N. A. Harron et al., "An image denoising model using deep learning for digital breast tomosynthesis images," in *IEEE 13th Control and Syst. Graduate Res. Colloq. (ICSGRC)*, IEEE, pp. 1–6 (2022).
29. M. Buda et al., "Detection of masses and architectural distortions in digital breast tomosynthesis: a publicly available dataset of 5,060 patients and a deep learning model," https://doi.org/10.1001/jamanetworkopen.2021.19100 (2020).
30. M. Juneja et al., "Denoising of magnetic resonance imaging using bayes shrinkage based fused wavelet transform and autoencoder based deep learning approach," *Biomed. Signal Process. Control* **69**, 102844 (2021).
31. D. Martin et al., "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. Eighth IEEE Int. Conf. Comput. Vision. ICCV 2001*, IEEE, Vol. 2, pp. 416–423 (2001).
32. A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.* **4**(2), 490–530 (2005).
33. D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory* **41**(3), 613–627 (1995).
34. D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 9446–9454 (2018).
35. M. Mirza and S. Osindero, "Conditional generative adversarial nets," https://doi.org/10.48550/arXiv.1411.1784 (2014).
36. J. Lehtinen et al., "Noise2noise: learning image restoration without clean data," https://proceedings.mlr.press/v80/lehtinen18a.html (2018).
37. F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychol. Rev.* **65**(6), 386 (1958).
38. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press (2016).
39. C. R. Harris et al., "Array programming with numpy," *Nature* **585**(7825), 357–362 (2020).
40. P. Virtanen et al., "Scipy 1.0: fundamental algorithms for scientific computing in Python," *Nat. Methods* **17**(3), 261–272 (2020).
41. S. Van der Walt et al., "Scikit-image: image processing in python," *PeerJ* **2**, e453 (2014).
42. M. Abadi et al., "TensorFlow: large-scale machine learning on heterogeneous systems," https://www.tensorflow.org/ (2015).
43. A. Paszke et al., "Pytorch: an imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst. 32* (2019).
44. Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process Mag.* **26**(1), 98–117 (2009).
45. D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika* **81**(3), 425–455 (1994).
46. H. Zhao et al., "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imaging* **3**(1), 47–57 (2016).
47. Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Thrity-Seventh Asilomar Conf. Signals, Syst. & Comput.*, IEEE, Vol. 2, pp. 1398–1402 (2003).
48. L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008).

Darlan M. N. de Araújo received his master's degree in 2021 and his computer science degree from the São Paulo State University (UNESP) in 2019. He has experience in the area of computer science, with emphasis on pattern recognition, digital image processing, computational vision and deep learning, working mainly in the following subjects: image noise filtering, medical imaging, and digital breast tomosynthesis.

Denis H. P. Salvadeo received his doctorate in 2013, his master's degree in 2009, and his degree in computer science from the Federal University of São Carlos in 2007. He is currently an assistant professor in the Department of Statistics, Applied Mathematics and Computation at UNESP, Campus of Rio Claro. He has experience in the areas of digital image processing, computer vision, and machine learning.

Davi D. de Paula received his master's degree in 2020 and his computer science degree from the UNESP in 2017. He is currently pursuing his PhD at the same university. He has experience in the area of computer science, with emphasis on image processing, machine learning, and deep learning.