Denoising PET Images With Neural Networks

HL2029 Medical Engineering Advanced Course

Andrés Martínez Mora [*]

August 2019

[*] Supervised by Massimiliano Colarieti-Tosti

Contents

1 Introduction

1.1 Positron Emission Tomography (PET)

Positron Emission Tomography or PET is a 3D imaging technique. It consists in the injection of radioisotopes that emit positrons or $\beta+$ particles which track some biomolecule of interest in the body. The positrons travel through a free path until encountering an electron. When doing so, two (approximately) opposite gamma rays are emitted. These rays are then registered in a radiation-sensitive detectors, so that by knowing the two points in the detector where both photons have arrived, one can know their complete trajectory, in what is called a Line Of Response or LOR. This technique is known as “electronic collimation” and avoids the use of plaques that allow photons just to come in one trajectory, known as “collimators”, and that reduce sensitivity.

PET detectors are usually composed by a scintillator crystal that transforms the gamma photons into visible light photons, registering in a sensitive device to light called Photomultiplier Tube or PMT. Normally, the scintillator crystals are divided into discrete elements, so that the size of the element tends to determine the spatial resolution of the image. The PMT is able to register the location of the event and produce a voltage proportional to the amount of produced light photons and to the amount of gamma photons that hit the detector. The voltage signal is further analyzed in terms of energy, so to distinguish “real” events from events that may have been deviated with scattering or coincidential events.

The usual geometry of a PET scan is the ring geometry. In here, the detector surrounds the patient, what helps to increase the amount of information. In order to increase this amount of information in the Z direction, several axial rings can be placed one after the other. Data acquisition is completed in two possible ways: by just acquiring the events that happen in the same ring for the same Z level or by allowing to count events happening between two different Z rings. The first acquisition type (direct) is later on simpler to reconstruct, but also misses much more events than the second type (transverse). In order to control the time decay of the radioisotope, the acquired events are listed and the time when they arrive is written down, so that the data is adjusted according to the decay of the radioisotope.

PET images tend to have a low sensitivity in comparison to other tomographic modalities as Computed Tomography (CT), since much less events are recorded, so it is very important to attempt to reduce image noise as much as possible in order to have the best image quality available.

1.2 Maximum Likelihood Expectation-Maximization (MLEM) reconstruction

MLEM is an iterative reconstruction technique for tomographic imaging, since the data acquired in tomography is not directly an image, but some information in an alternative space. The raw data appear as "sinograms", where all the spatial information is projected into a line with a forward operator, the Radon Transform, in a specific direction, assuming that the sinogram is the sum (or line integral, to be more specific) of all the points of the volume in that direction. Consequently, by combining all these lines from different directions, one has the tomographic image codified as a sinogram. The general equation for the tomographic reconstruction process is defined as:

$$Ax + n = b \tag{1}$$

Where "A" is the forward operator, "x" is the real volume, "b" is the sinogram and "n" is some noise quantity that will always be present during acquisition. In the reconstruction process, "x" is desired to be obtained from "b". However, this is an ill-posed problem, as the noise quantity is stochastic, so different reconstruction methods try to approximate "x", but never get to the real value, having a trade-off between accuracy and noise, so that the closer one is to the real solution, the noisier the results will be.

Starting from the raw data, MLEM takes an initial guess of how the reconstructed image could be. The first initial guess is no more than a backprojection of the raw data, which consists in the assumption that all the voxels in a specific direction kept by the sinogram have the same value as the value in the sinogram. The initial guess is then projected and compared to the original sinogram. According to this comparison, MLEM refines the way to provide the next initial guess in the following iteration, until reaching convergence. The method is quite fast, as with few iterations it reaches convergence. What is more, as the method tries to solve an ill-posed problem as tomographic reconstruction is, it can also lose accuracy as more iterations are applied, so neither is it recommended to apply many iterations.

1.3 Convolutional Neural Networks (CNNs)

Neural Networks (NNs) are Machine Learning algorithms involved in supervised regression or classification problems where the data used to build the network, known as training data, is labelled. In other words, the class or value of the training data is known. In a NN, the input data is processed in a series of "neurons". A "neuron" in this context is an operator that receives a set of weights from all inputs and optionally a bias and which performs a linear combination of those weights and biases that influence the inputs. The result of this operation is fed into a function called “activation function”. In many cases, the activation function is a Rectified Linear Unit or ReLU, which is zero if the linear combination of weights, inputs and biases is less or equal than zero and which keeps the result of the linear combination if the result is positive. Neurons can be combined either at the same time, analyzing the inputs with different weights and biases, or sequentially, one after other. In the first case, the neurons are located in the same layer, while in the second case, the neurons are located in sequential layers. The combination of all the neurons in different layers is called “network” and its main goal is to perform regression or classification with the adequate weights and biases.

The process where the network’s weights and biases are tuned to solve a specific problem is known as training, being usually done with a method called “backpropagation”. In backpropagation, a series of initial weights and biases is proposed, so that some input data are introduced into the network and since the learning is supervised, one knows the real output or “target” of the network, so one can compute an error function (also known as cost function or loss function) by comparing the current output with the target. Then, the output is propagated backwards through the network, in order to look for those weights and biases that minimize the error function. Optimization methods to do this are Stochastic Gradient Descent (SGD), although other optimizers as Adam are also popular.

Some techniques to make the network more efficient are for example batch normalization. In batch normalization, the mean and standard deviation of the batch (set of data used at once) used to train the network are computed, so that the mean is subtracted to the input batch and the result is divided by the standard deviation. This allows to make the network much more robust, as all features to be analyzed in the network are studied under the same scale. It can also help to increase the network speed, too.

There are special types of NNs suited for different kinds of input data. This is the case of Convolutional Neural Networks (CNNs), where the input data is processed in each neuron with convolutions. This allows to exploit the quality of input data that has close relationships to one another, as happens in images, being of high importance in image processing. After performing convolutions to the data, the image is resampled to a lower resolution and increased in the number of analyzed channels, in a process called “pooling”. These channels are equivalent to the number of neurons per layer in a classical NN. As a consequence of this, CNNs are mainly made of convolutional and pooling layers one after the other. Classical

CNNs contain a last layer where the results are processed with one-dimensional linear combinations, called “softmax”. Nevertheless, some recent CNN architectures lack this layer, being known as “fully convolutional NNs”. Some other architectures are made of a downsampling path, where the resolution is decreased to perform some operation, and then incorporate an upsampling path, where the resolution is recovered to the original values. This architecture is known as ”U-Net”. In summary, the main goal of CNN training is to find those kernel coefficients that minimize the error function. CNNs can be improved with batch normalization, too.

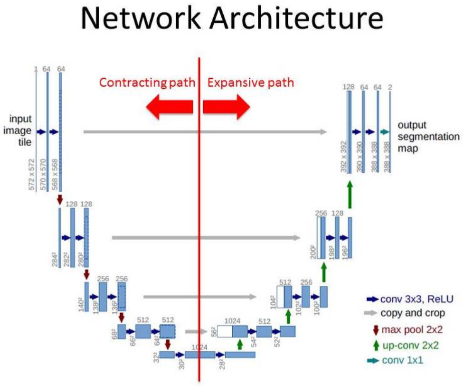

Figure 1: U-Net: general scheme

1.4 Figures of Merit (FOMs)

NNs can be used for denoising images, being compared to some ground truth (if this is available). The methods used to compare the denoising results to the ground truths are called Figures of Merit (FOMs). Some examples of FOMs are the *Structural Similarity Index* or SSIM or the *Peak Signal-to-Noise Ratio* or PSNR. The SSIM measures how “good” is the quality of the denoised image by comparing it to a ground truth. The mean, standard deviation and covariance of voxel values in windows of the ground truth and the denoised image are computed and combined to calculate the SSIM with the following equation:

$$SSIM = \frac{(2\mu_{ground-truth} \cdot \mu_{denoised} + c_1) \cdot (2\sigma_{ground-truth,denoised} \cdot \mu_{denoised} + c_2)}{(\mu^2_{ground-truth} + \mu^2_{denoised} + c_1) \cdot (\sigma^2_{ground-truth} + \sigma^2_{denoised} + c_2)} \quad (2)$$

Where μ is the mean value of the ground truth or the denoised image, $\sigma_{ground-truth,denoised}$ is the covariance of the ground truth and the denoised image and σ^2 is the variance of the ground truth or the denoised image. The SSIM takes values between 0 and 1, so that the result is better as the SSIM approaches to 1.

The PSNR also establishes a comparison between the ground truth and the denoised, measuring the maximum possible power of the image signal over the noise quantity in the image. As the image is more denoised, it is supposed to have a larger PSNR. The PSNR relies on the Mean Square Error (MSE) between the denoised image and the ground truth, being computed in the following way:

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX}{\sqrt{MSE_{ground-truth,denoised}}}\right)(dB) \tag{3}$$

The PSNR is measured in dB, so that "MAX" is the maximum value that a voxel can take in the images, usually $2^n - 1$, where "n" is the number of bits codifying each voxel value and "MSE" is the Mean Square Error between the ground truth and the denoised image.

2 Objective

Until quite recently, PET images reconstructed with MLEM algorithm have been denoised with CNNs that have been trained with a set of images from MNIST database, which consists on images of handwritten digits. Then, it is desired to see if by training the same CNN with images with more general shapes, rather than handwritten numbers, as could be ellipsoids, the performance is enhanced or not, with respect to state-of-the-art methods, as are MLEM reconstructions of about ten iterations.

3 Materials & Methods

3.1 Simulations with ODL® phantoms

The first step consisted in working *in silico* with phantoms available in a computational software package called ODL® (Operational Discretization Library).

The space and geometry for the phantoms were defined with ODLPET library, a special package for PET images. Two types of 2D geometries were used: one with a simple resolution of 28x28 and another with the actual geometry of the mini-PET scanner from KTH laboratory. The 3D geometry used was always the one from the mini-PET scanner.

With the given spaces and geometries, a function to produce phantom images with random ellipsoids was elaborated. There have been similar previous implementations, but they left the final phantom with a high concentration of ellipsoids in the center of the image, what could be translated in the real world as an unreal excess of radioisotope activity in the center of the phantom. Thus, interest was placed in designing phantom images where the centers of the different ellipsoids could be placed outside the Field Of View (FOV) of the image, so that the concentration of ellipsoids in the center was much lower and much more realistic. The task was completed by creating a "surrounding" space around the defined space, placing the ellipsoids there and then cropping the central space with the size of the originally defined space. Then, random calls for the ellipsoids' intensities, centers, axes lengths and angulations were completed for a certain number of random ellipsoids. This function was implemented to work both in 2D and 3D, being called as many times as the number of images desired for training and testing.

The next step was to compute the reconstruction of the projected sinograms with the MLEM algorithm. Two different reconstructions were completed: one with just one iteration that was equivalent to a backprojection and that was the input to the later denoising network, and another one with an optimal number of iterations (usually, 10 iterations), that was used as a “simulated state-of-the-art” to compare it to the denoising results that were obtained later.

CNNs were used to denoise the noisy reconstructed images with only one MLEM iteration. The architecture chosen was the same for 2D and 3D images, working with an extra dimension in the 3D case. It was an ”U-Net”, which optimally used three downsampling and upsampling layers in the 2D case and an extra layer for the 3D case.

The networks were trained with two datasets: one with random ellipsoids (generated as explained three paragraphs above) and another with MNIST handwritten digit images, being both compared to the state of the art in PET image reconstruction: a 10 iteration MLEM reconstruction. The validation was always done with random ellipsoid images. In 2D images, the networks were trained with groups of 50 images, known as batches, while they were evaluated with an only testing image. For the 3D case, as the images occupied a larger space in memory, the training batches contained just a downsampled volume, while the testing volumes were also downsampled. During the project, as the 3D images were not providing the desired results, the networks not only received as inputs images with random ellipsoids, but also received empty images and images with very few ellipsoids, so to increase the variability of the inputs introduced to the network. In addition, as the results from the MNIST network with 2D images was not so good, it was discarded to use this network on 3D images, where the complexity wa higher.

The networks were evaluated with a Smooth L1 cost function, which computed a quadratic error for low values and an absolute error for high values. The lower the error was, the closer the denoised images were to the ground truth. The optimizer used for the networks was Adam, using a learning step of 1.5e-3. Training and validation costs from the Smooth L1 operator were saved to elaborate learning curves. The number of iterations for training the network, known as *epochs* was of ten for the 2D images and just of one for the 3D images, as it was very lengthy, in computational terms. In case it was needed to do a further training, the model amd the optimizer parameters were saved and the training was then started from these pre-computed parameters. By saving these parameters, three extra epochs were executed in the network trained with 2D ellipsoid images with the mini-PET resolution, while the network trained with 3D ellipsoid volumes was fed with more training volumes.

The denoising process was evaluated in terms of the FOMs presented in the Introduction section: SSIM and PSNR. Additionally, the results from the images were also interpreted in 1D, obtaining the central profiles of the images.

4 Results

4.1 Simulations with ODL® phantoms

4.1.1 Low resolution 2D images

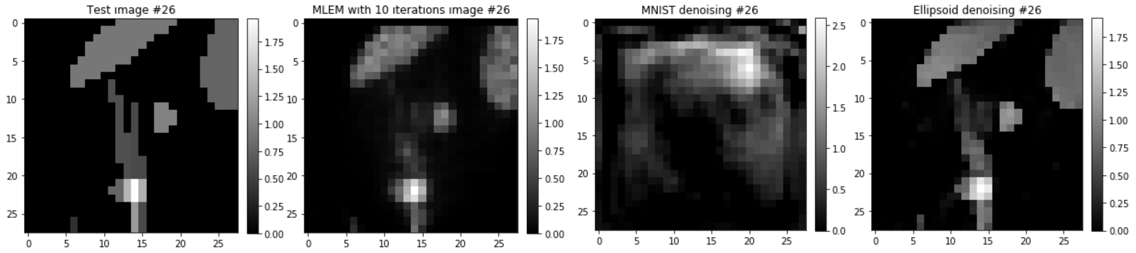

Figure 2: Low resolution 2D images. Left*: ground truth /* Center left*: 10 iteration MLEM reconstruction* Center right*: Denoised with MNIST database* Right*: Denoised with ellipsoid database*

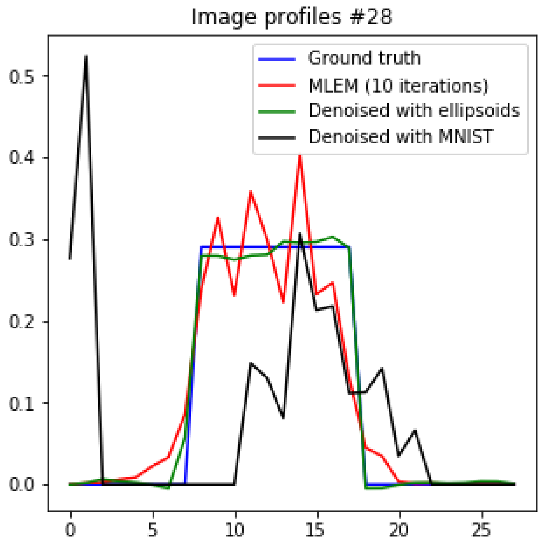

Figure 3: Center Profiles of the 2D low resolution images. Blue*: Ground Truth,* Red*: 10 iteration MLEM,* Black*: MNIST denoising,* Green*: Ellipsoid denoising*

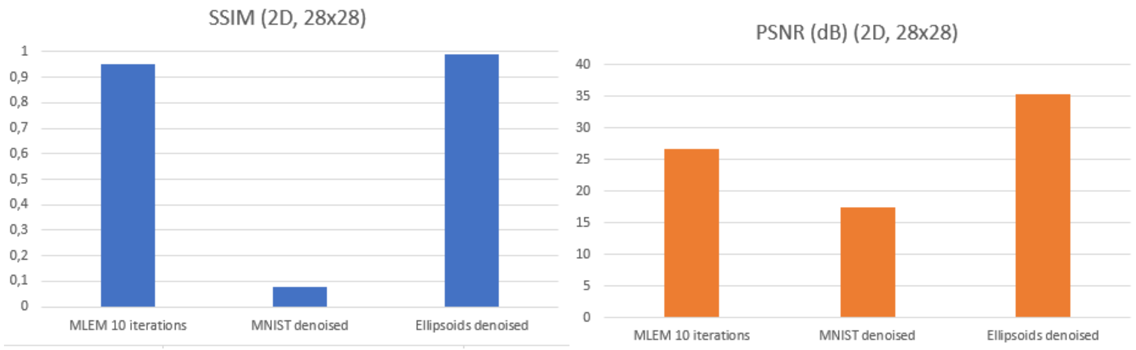

Figure 4: FOM values for low resolution 2D images. Left*: SSIM for the 10 iterations MLEM reconstruction, the MNIST denoising and the ellipsoid denoising, respectively /* Right*: PSNR for the 10 iterations MLEM reconstruction, the MNIST denoising and the ellipsoid denoising, respectively*

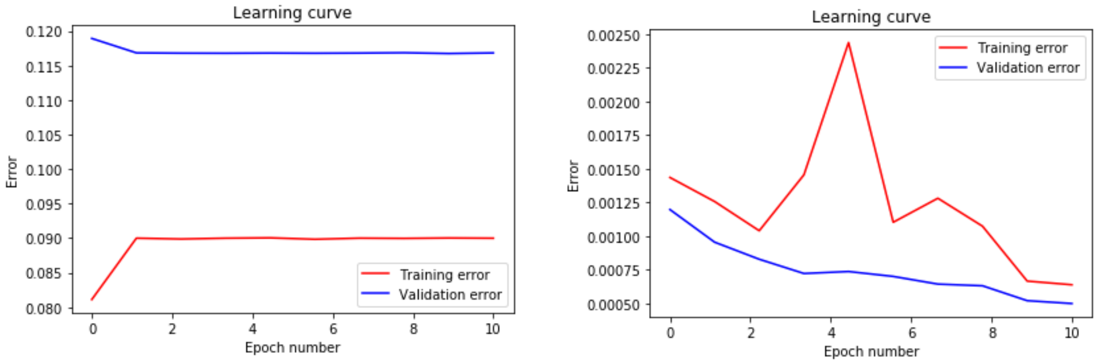

Figure 5: Learning curves for low resolution 2D images. Training errors are in red, while validation errors are in blue. Left*: Learning curve for the network working with MNIST images /* Right*: Learning curve for the network working with ellipsoid images*

4.1.2 High resolution 2D images

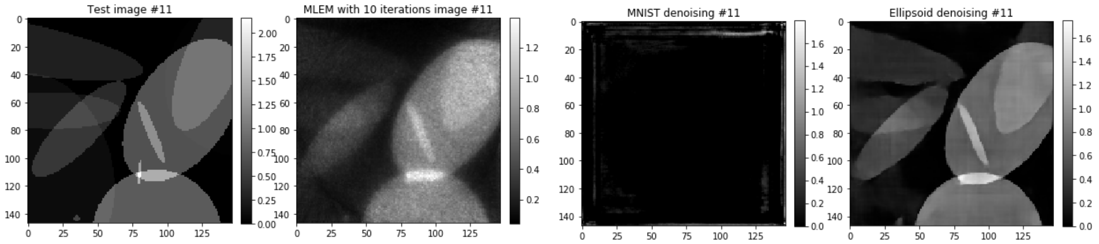

Figure 6: High resolution 2D images. Left*: ground truth /* Center left*: 10 iteration MLEM reconstruction* Center right*: Denoised with MNIST database* Right*: Denoised with ellipsoid database*

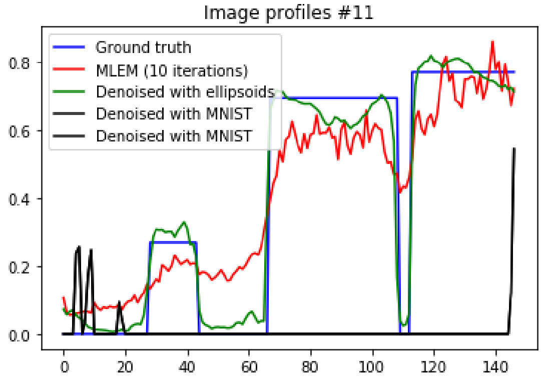

Figure 7: Center Profiles of the 2D high resolution images. Blue*: Ground Truth,* Red*: 10 iteration MLEM,* Black*: MNIST denoising,* Green*: Ellipsoid denoising*

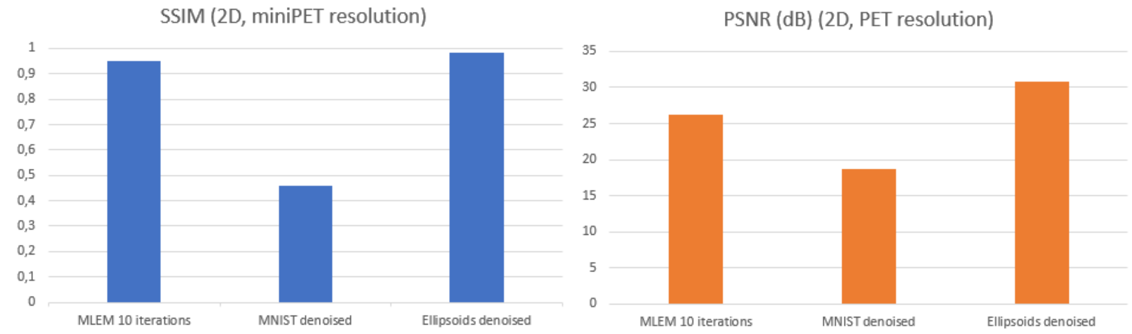

Figure 8: FOM values for high resolution 2D images. Left*: SSIM for the 10 iterations MLEM reconstruction, the MNIST denoising and the ellipsoid denoising, respectively /* Right*: PSNR for the 10 iterations MLEM reconstruction, the MNIST denoising and the ellipsoid denoising, respectively*

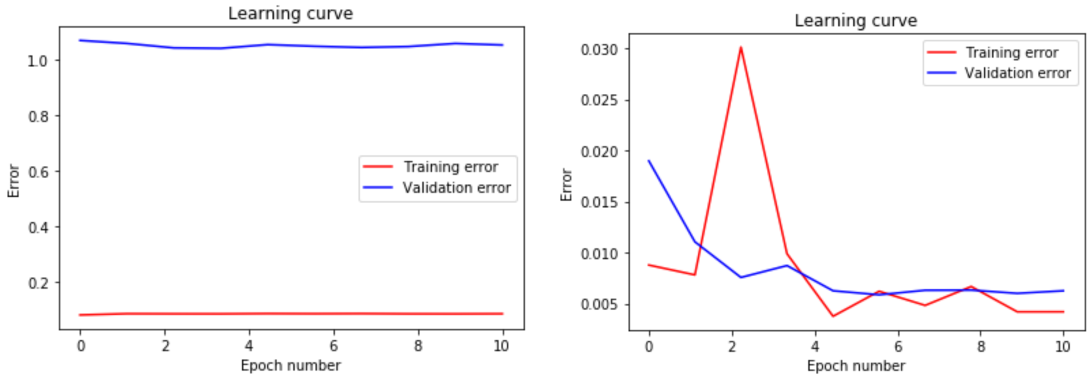

Figure 9: Learning curves for low resolution 2D images. Training errors are in red, while validation errors are in blue. Left*: Learning curve for the network working with MNIST images /* Right*: Learning curve for the network working with ellipsoid images*

4.1.3 3D images

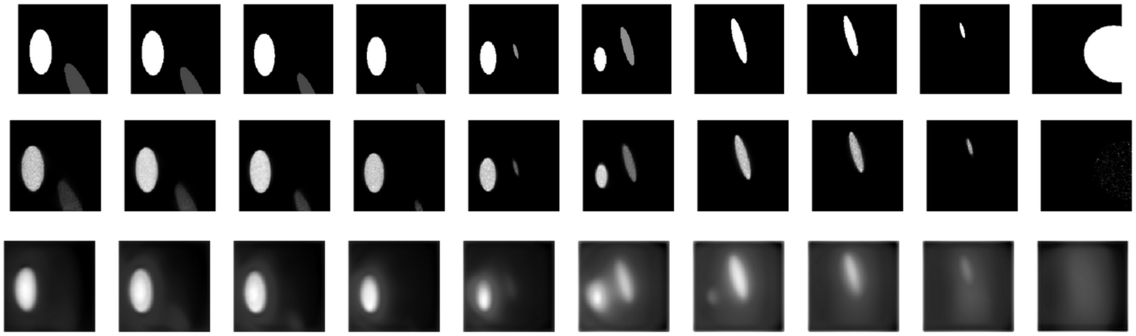

Figure 10: 3D fly through synthetic images in the axial direction. First row*: Ground truth /* Second row*: 10 iterations MLEM reconstruction /* Third row*: ellipsoid denoised*

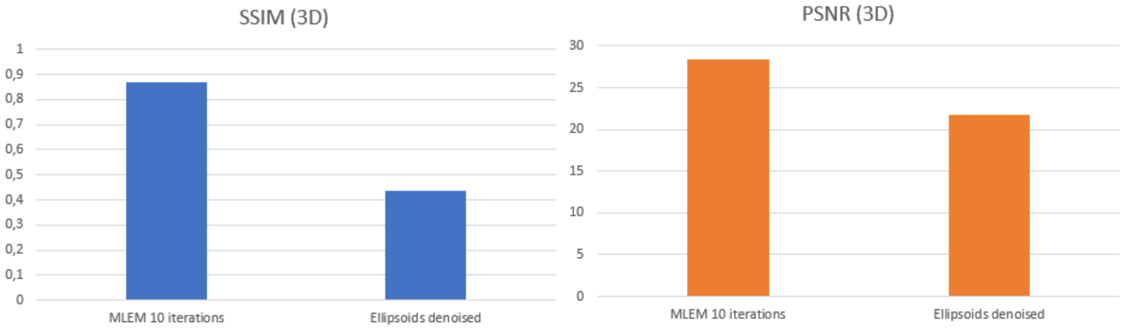

Figure 11: FOM values for 3D images. Left*: SSIM for the 10 iterations MLEM reconstruction and the ellipsoid denoising, respectively /* Right*: PSNR for the 10 iterations MLEM reconstruction and the ellipsoid denoising, respectively*

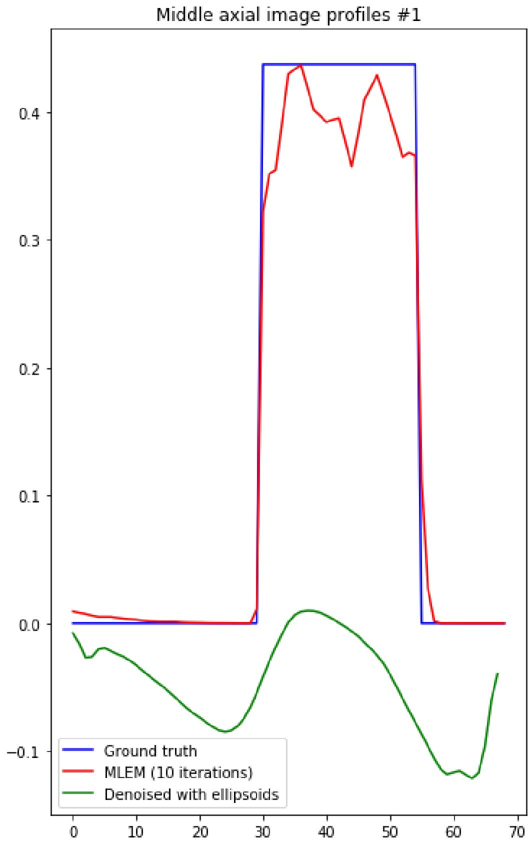

Figure 12: Center Profiles of the 3D images in the axial direction. Blue*: Ground Truth,* Red*: 10 iteration MLEM,* Green*: Ellipsoid denoising*

5 Discussion

In the 2D case with low resolution images, the newly proposed network that was trained with ellipsoids provides better results than the state-of-the-art (the 10 iterations MLEM reconstruction) and much better results than the denoising with the MNIST database. This can be observed in the image comparison carried out in Figure 2, where the interior of the ellipsoids is not so abrupt in the resulting images from the ellipsoid network as in the 10 iterations MLEM image. In addition, the central profile from the ellipsoid network, which is the green line in Figure 3, falls much closer to the profile of the ground truth (in blue) than the profile of the 10 iterations MLEM reconstruction (in red). In Figure 4, the values of SSIM and PSNR are higher for the ellipsoid network than for the 10 iterations MLEM reconstruction. The same values for the case of the MNIST network are much lower. What is more, the resulting images did not contain any kind of artifact, so that the ellipsoids network was just focused on denoising and did not distort the images. All these results on "basic" data were quite promising and encouraged to test the same algorithm on images with the resolution of the mini-PET scanner.

With respect to the learning curves of the networks used to denoise the low resolution 2D images, the errors decrease with the epoch number, something expected. However, something unusual happened in the learning curve for the ellipsoid network (the new one), as the training error is in all epochs higher than the validation error. It does not make sense that the network performs better on images that it has not learned than on images that it has been learning. It has to be acknowledged that training with just ten epochs may not be enough, so it could happen that if the network had been further trained, the error tendency could have been inverted. As it was expected, the errors in the network trained with the MNIST database are much higher than in the network trained with the ellipsoids.

For the case of 2D images with a proper resolution for a regular scanner, the proposed network kept performing better than the state-of-the-art, but not in such an enhanced way as in the low resolution case. In here, the images and central profiles are more similar, as can be observed in Figures 6 and 7, respectively. Unlike what happened in the low resolution images, it seems that the NNs distort a little bit the images, especially in the interior of the ellipsoids, being this the reason why the enhancement is not as good as in the low resolution case. This makes that the SSIM and PSNR values for the ellipsoid network are much closer to the state-of-the-art, having a lower enhancement than in the low resolution images.

With respect to the learning curves of both networks (Figure 9), in this case there is no inversion of error values between training and validation, although the training error curve for the ellipsoid network contains many fluctuations. It is important to remark that the learning curve values for training and validation in the MNIST network are quite different and hardly decrease, staying almost constant throughout the epochs, what shows the inefficiency of the MNIST network to denoise images with other objects than handwritten numbers.

In the case of the 3D images, the results are not so promising as in the 2D cases. In here, the the results provided by the ellipsoid network are worse than the ones given by the state-of-the-art, both in quantitative and qualitative terms. About the image appearance (Figure 10), the NN seems to do some denoising, but it seems to be incomplete, like an MLEM reconstruction with few iterations, and if one has a look to the last image to the right, there should be a big white circle that is lacking. Consequently, this decreases the FOM values, as can be observed in Figure 11, and the denoised profiles are far away from those of the ground truth and the state-of-the-art (figure 12).

One of the main barriers to produce better results than the ones achieved, especially in the 3D case, has been the computational power. Perhaps the use of a better computational hardware could have provided better quality results, so it may be considered to shift to more powerful computers in terms of parallel processing.

6 Conclusion & Future Work

The proposed method of denoising and removing reconstruction artifacts with a CNN that receives more general shapes than handwritten numbers seems highly promising for 2D synthetic images, as it provides images with a better appearance and with a higher SSIM and PSNR when compared to a ground truth. However, some minor artifacts in the interior of highly homogenous surfaces must be also taken into account. Nevertheless, when working with 3D volumes, the proposed algorithm does not seem to be as effective as in 2D surfaces, leaving many undesired residual artifacts, what makes that the PSNR and SSIM values are not as high as in the 2D case. It has to be acknowledged that the computational cost and time required to process 3D volumes was an important barrier in order to try to fix the not so good performance of the 3D image denoising.

The network trained with the MNIST database provided very bad results. However, this is not a matter of worry, as the results from this network were just obtained to show that a network trained with more general shapes performs much better than a network that is only trained with handwritten digits.

As for future steps in this work, I would consider to test the algorithm that works with 2D images with a larger quantity of epochs in order to try to reduce the artifacts that the network causes in the interior of homogeneous surfaces. However, most of the focus should be placed in the algorithm working with 3D images, as the results at the moment are not so promising as they are with the 2D images. I would suggest to train with larger datasets, with more epochs and with higher data variability, even considering the development of alternative network architectures or changing the number of layers of the current U-Net architecture. Once the performance of the algorithm for the 3D volumes could be enhanced, the next thing to do would be to work with realistic data, both for training and for testing, where the proposed method would arrive to its crucial moment: determining if it is also able or not to denoise real images with a NN based on general shapes. The access to more powerful hardware could answer this question in the lowest time possible.

List of Figures

List of Acronyms

CNN: Convolutional Neural Network
CT: Computed Tomography
CV: Coefficient of Variation
FDG: FluDeoxyGlucose
FOV: Field Of View
FOM: Figure Of Merit
LOR: Line Of Response
MLEM: Maximum Likelihood Expectation-Maximization
MSE: Mean Square Error
NN: Neural Network
ODL®: Operational Discretization Library®
ODLPET®: Operational Discretization Library® for Positron Emission Tomography
PET: Positron Emission Tomography
PMT: PhotoMultiplier Tube
PSNR: Peak Signal-to-Noise Ratio
ReLU: Rectified Linear Unit
ROI: Region Of Interest
SGD: Stochastic Gradient Descent
SSIM: Structural Similarity Index