2019 27th European Signal Processing Conference (EUSIPCO)

Hyperspectral X-ray Denoising: Model-Based and Data-Driven Solutions

Nicolò Bonettini, Marco Paracchini, Paolo Bestagini, Marco Marcon, Stefano Tubaro
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133, Milano, Italy

***Abstract*—In this paper we deal with the problem of hyperspectral X-Ray image denoising. In particular, we compare a classical model-based Wiener filter solution with a data-driven methodology based on a Convolutional Autoencoder. A challenging aspect is related to the specific kind of 2D signal we are processing: it presents mixed dimensions information since on the vertical axis there is the pixels position while, on the abscissa, there are the different wavelengths associated to the acquired X-Ray spectrum. The goal is to approximate the denoising function using a learning-from-data approach and to verify its capability to emulate the Wiener filter using a much less demanding approach in terms of signal and noise statistical knowledge. We show that, after training, the CNN is able to properly restore the 2D signal with results very close to the Wiener filter, honouring the proper signal shape.**

***Index Terms*—Hyperspectral Imaging, Image Denoising, Convolutional Autoencoder, Machine Vision.**

I. Introduction

Wiener filter [1] is one of the milestones in signal and image denoising and still nowadays is a crucial stage in state-of-the-art image denoising [2] and image processing [3] techniques. However, the introduction of Deep Learning dramatically outperformed most of the previous approaches making Convolutional Neural Networks (CNN) a paramount step in many denoising algorithms [4]–[6]. In this article we tackle the denoising of mixed 2D signals, where, as detailed in section II, we have different physical characteristics on the two axes (space and X-ray energies in our case). In particular we compare the capabilities of the 2D Wiener filter to a CNN approach with the significant advantage that, for the latter, we do not need a detailed knowledge of noise and signal statistical behaviour. We follow an approach based on a Convolutional Autoencoder due to its proved effectiveness in many denoising applications [5], [7]–[10] and the results provide a very good agreement with the Wiener filter. The rest of the paper is organized as follow: in Sec.II the hyperspectral denoising problem is defined and the acquisition system is described. In Sec.III the two proposed solutions are described, including some preprocessing steps. In Sec.IV the experimental setup used to collect hyperspectral data is described. Furthermore, in Sec.V the denoising algorithms described in Sec.IV are applied and the methods are compared. Finally in Sec.VI the conclusions are drawn including some ideas for further development.

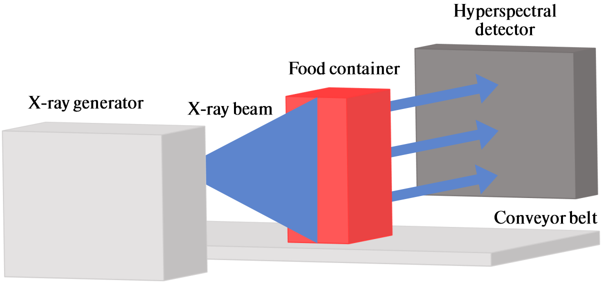


Fig. 1: Acquisition system.

II. Background and Problem Statement

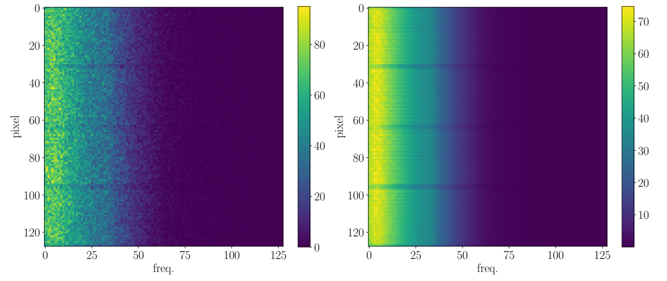
In this section, we briefly describe the image acquisition system and we formally define the image denoising problem. We then provide some additional background remarks on the used methodologies, which might be useful to understand the rest of the paper.

Image acquisition system. With reference to Fig. 1, the hyperspectral image acquisition system considered in this work is composed by a X-ray generator, a conveyor transporting the objects under analysis (e.g., food containers) and a hyperspectral X-ray detector. The source of the X radiation is an X-ray tube, i.e., a vacuum tube containing a cathode and an anode. X-rays are generated by directing a stream of high speed electrons from the cathode to the anode. The detector is a linear sensor that measures the intensities (at different spectral frequencies, commonly named X-Ray energies due to the Planck-Einsten Relation [11]) of photons that have not been absorbed by objects in front of it, i.e., the photons that pass through the object reaching the sensor.

The acquired spectrum is divided into W photons frequencies intervals, also called energy bins. Therefore, each pixel of the acquired image is a vector of size $(1 \times W)$ containing the intensities at all the energy bins. Since the detector is a push broom sensor of H vertically aligned pixels, the acquired 2D signal is a linear hyperspectral image $\mathbf{I}$ (called image in the rest of the paper) with the size of $(H \times W)$ samples, and each value $[\mathbf{I}]_{ij}$, $i = \{1, \dots, H\}$, $j = \{1, \dots, W\}$ represents the intensity of received photons for a given spatial position and a given energy bin.

This research was partially supported by Xnext® s.r.l. who provided the complete data acquisition system together with insight and expertise that greatly assisted the research.

978-9-0827-9703-9/19/$31.00 ©2019 IEEE



(a) (b)

Fig. 2: (a) Single acquisition of background image. (b) Mean of 10000 acquisitions of background image.

It is important to point out that, in normal operating condition of the system, the scanned object is moving on the conveyor belt. It is therefore paramount to develop timely and efficient solutions for data denoising and processing in order to work in real-time environments.

Problem formulation. Images acquired through digital acquisition systems are subject to different kinds of degradation. For the system under analysis, we can approximate noise as additive zero-mean noise due to thermal, shot and $1/f$ noise [12].

Formally we can define the acquired image $\mathbf{I}$ as:

$$\mathbf{I} = \mathbf{I}_0 + \mathbf{N}, \tag{1}$$

where $\mathbf{I}_0$ is the true (i.e., unaffected by noise) image signal and $\mathbf{N}$ is the additive noise term.

With this model at hand, our goal is to estimate a denoised version of $\mathbf{I}$, namely $\hat{\mathbf{I}}$, as close as possible to $\mathbf{I}_0$, thus compensating for the detrimental effect of $\mathbf{N}$.

Under the assumption of additive zero-mean independent noise realizations affecting the acquisition, we can consider the sample mean $\bar{\mathbf{I}}$ of an adequate number of repeated acquisitions of an image $\mathbf{I}$ as a good and not distorted estimate of the clean image $\mathbf{I}_0$. Therefore we consider $\bar{\mathbf{I}}$ as the ground truth for our algorithm's output. (see Fig. 2). However, note that the sample mean can only be applied in a controlled offline procedure, as it would not be feasible in a real-time system due to the necessity of multiple repeated acquisitions.

Wiener filter for image restoration. Wiener filtering is commonly adopted in image restoration problems, since it performs a statistical estimation of an unknown signal while taking noise into account. In particular, Wiener filter approach estimates $\hat{\mathbf{I}}$ such that the Mean Squared Error between $\mathbf{I}_0$ and $\hat{\mathbf{I}}$ is minimized as it follows:

$$[\hat{\mathbf{I}}]_{ij} = \mathcal{F}_2{}^{-1}\left([\mathcal{F}_2(\mathbf{I})]_{ij}\,\frac{[\mathbf{S}_I]_{ij}}{[\mathbf{S}_{I_0}]_{ij} + [\mathbf{S}_N]_{ij}}\right), \tag{2}$$

where $\mathcal{F}_2(\cdot)$ is the 2D Fourier transform, $\mathbf{S}_N = |\mathcal{F}_2(\mathbf{N})|^2$ represents the power spectrum of the noise, and $\mathbf{S}_I = |\mathcal{F}_2(\mathbf{I})|^2$ is the power spectrum of the clean image. This method is very effective provided that $\mathbf{S}_N$ and $\mathbf{S}_I$ are estimated in a quite accurate way and model's assumptions hold [1].

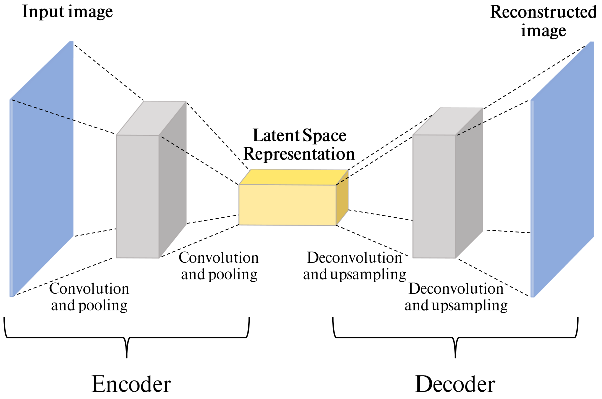


Fig. 3: Typical Autoencoder structure. An input image is mapped to its latent representation by the encoder. The decoder turns the latent representation into an estimate of the input data.

Convolutional Autoencoder. An Autoencoder is a specific kind of neural network. The purpose of such a network architecture is to learn a low-dimensionality representation of an image (latent representation), and then reconstruct the input from it [13]. The rationale behind this technique is to build a representation of the input such that the reconstructed output is derived from its most robust features. The Convolutional Autoencoder (Fig. 3) is composed by an encoding part, in which Convolution and Pooling layers are employed to reduce the input dimensionality, and a decoding part, in which Deconvolution (i.e. Transposed Convolution) and Upsampling layers are employed to expand the latent representation dimensionality up to the input shape. We can define the Convolutional Autoencoder as a function $\mathcal{A}(\boldsymbol{\theta}, \cdot)$ such that:

$$\hat{\mathbf{I}} = \mathcal{A}(\boldsymbol{\theta}, \mathbf{I}), \tag{3}$$

where $\mathbf{I}$ is the input image, $\hat{\mathbf{I}}$ is the output image, matching the shape of $\mathbf{I}$ and $\boldsymbol{\theta}$ is the Autoencoder weights vector that must be learned with a suitable training procedure.

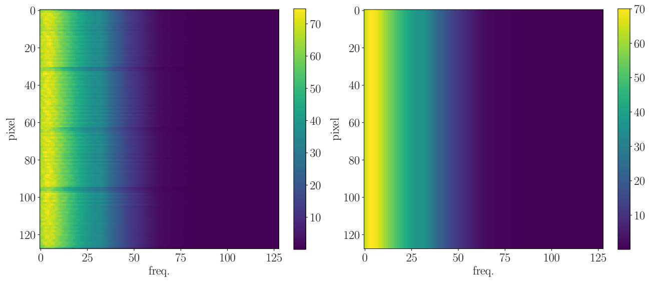
III. Algorithm

In this section we describe the used algorithm, which is composed of a preprocessing procedure followed by the actual denoising step. In the following we report a detailed description of each step.

A. *Preprocessing*

Images directly obtained from the sensor are not suitable for denoising due to their dynamic range and the presence of banding artifacts. For this reason, we first apply two preprocessing operations before performing the denoising step.

Row Normalization. Fig. 4a shows the average background image $\mathbf{M}$ obtained averaging 10 000 background images (no objects were placed in front of the sensor). It is possible to notice that the average acquired signal shows three horizontal lines in which the values are significantly lower than the rest



(a) (b)

Fig. 4: Mean of background image. (a) Before row normalization. (b) After row normalization.

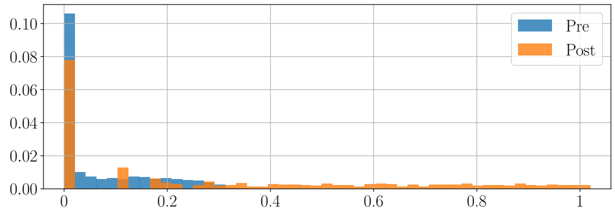


Fig. 5: Values distribution for an image pre and post histogram normalization.

of the signal's rows. This effect is due to the form factor of the detector which is physically divided in four sub-sensors, each one contributing to 32 rows. To contrast this, we perform row normalization on each $H \times W$ input image $\mathbf{I}$, starting from the mean background signal $\mathbf{M}$. The background signal is chosen since all the other acquired images have lower values (in the background image no objects intersect the X-rays path so the intensity of the detected photons is maximum).

The normalized version $\tilde{\mathbf{I}}$ of the input image $\mathbf{I}$ is obtained element-wise by:

$$[\tilde{\mathbf{I}}]_{ij} = \frac{[\mathbf{I}]_{ij}\,[\mathbf{m}]_j}{[\mathbf{M}]_{ij}}. \tag{4}$$

where:

$$[\mathbf{m}]_j = \frac{1}{H}\sum_{i=1}^{H}[\mathbf{M}]_{ij}, \tag{5}$$

is the spatial average of $\mathbf{M}$ for frequency bin j. Fig. 4b reports the normalized image $\tilde{\mathbf{I}}$. It is possible to notice that the periodic artifacts have been strongly attenuated.

Histogram Normalization. Another important preprocessing step consist in stretching the image dynamic in the range $[0, 1]$ in order to obtain an adequate input for the CNN. Since we are dealing with hyperspectral data, different images have different value distributions with their own maxima. We perform histogram normalization in order to obtain a more uniform distribution of data over the range $[0, 1]$ as shown in Fig. 5. This is convenient for numerical stability in optimization algorithms, since it is more difficult reaching convergence with very skewed and imbalanced sample distributions.

TABLE I: Convolutional Autoencoder layers.

Type	**Filters**	**Activation**	**Output shape**
Input	-	-	$(128 \times 128 \times 1)$
Convolution 2D	$32 \times (5 \times 5)$	ReLU	$(124 \times 124 \times 32)$
Max Pooling	(2×2)	-	$(62 \times 62 \times 32)$
Convolution 2D	$64 \times (3 \times 3)$	ReLU	$(60 \times 60 \times 64)$
Max Pooling	(2×2)	-	$(30 \times 30 \times 64)$
Convolution 2D	$128 \times (3 \times 3)$	ReLU	$(28 \times 28 \times 128)$
Max Pooling	(2×2)	-	$(14 \times 14 \times 128)$
Convolution 2D	$256 \times (3 \times 3)$	ReLU	$(12 \times 12 \times 256)$
Max Pooling	(2×2)	-	$(6 \times 6 \times 256)$
Upsampling	(2×2)	-	$(12 \times 12 \times 256)$
Deconvolution 2D	$128 \times (3 \times 3)$	ReLU	$(14 \times 14 \times 128)$
Upsampling	(2×2)	-	$(28 \times 28 \times 128)$
Deconvolution 2D	$64 \times (3 \times 3)$	ReLU	$(30 \times 30 \times 64)$
Upsampling	(2×2)	-	$(60 \times 60 \times 64)$
Deconvolution 2D	$32 \times (3 \times 3)$	ReLU	$(62 \times 62 \times 32)$
Upsampling	(2×2)	-	$(124 \times 124 \times 32)$
Deconvolution 2D	$1 \times (5 \times 5)$	SoftMax	$(128 \times 128 \times 1)$

B. *Denoising*

After preprocessing, images are ready for the denoising step. In this work, we compare two different denoising strategies: a model-based solution based on Wiener filtering; a data-driven method based on autoencoders.

Wiener Filter. According to (2), in order to apply Wiener filter to an image, we need an estimate of the power spectrum of the clean image $\mathbf{S}_I$ and the power spectrum of the noise $\mathbf{S}_N$. The power spectrum of the clean image $\mathbf{S}_I$ is computed as $\mathbf{S}_I = |\mathcal{F}_2(\bar{\mathbf{I}})|^2$ since the normalized mean image $\bar{\mathbf{I}}$ is considered representative of the clean image, where:

$$[\bar{\mathbf{I}}]_{ij} = \frac{1}{10000}\sum\nolimits_{n=1}^{10000}[\tilde{\mathbf{I}}_n]_{ij}, \tag{6}$$

Following the same procedure, the power spectrum of the noise $\mathbf{S}_N$ is computed as $\mathbf{S}_N = \frac{1}{10000}\sum_{n=1}^{10000}|\mathcal{F}_2(\mathbf{I}_n - \bar{\mathbf{I}})|^2$.

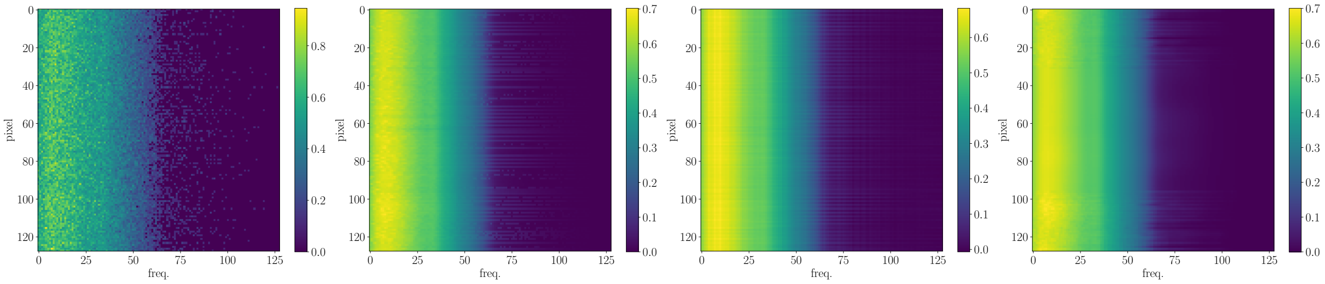
Convolutional Autoencoder. The detailed architecture of the adopted Convolutional Autoencoder is depicted in Table I. Given as input a noisy image $\mathbf{I}$, the Convolutional Autoencoder $\mathcal{A}(\boldsymbol{\theta}, \mathbf{I})$ outputs directly the denoised version of the image $\hat{\mathbf{I}}$. The goal of the autoencoder is to first reduce the dimensions of the image and then try to reconstruct it from this low-dimension representation, hopefully neglecting the noise affecting the original signal. The proposed architecture takes as input an image of shape $128 \times 128 \times 1$ and reduces it to $6 \times 6 \times 256$, applying a downsampling factor of almost 2. The use of SoftMax as last layer's activation function ensures that the output lies in the $[0, 1]$ range.

IV. Experimental Setup

In this section we report all details concerning the considered experimental setup. We first describe the considered dataset. We then report implementation details about the autoencoder training procedure.

Dataset. The dataset is composed by acquisitions taken with MultiX ME100[1], which is a linear sensor of length 128 pixel.

[1] https://www.qualityassurancemag.com/article/multix-me100-x-ray/



(a) Input (b) Ground truth (c) Wiener filter (d) Convolutional Autoencoder

Fig. 6: Output images from the considered denoising methods.

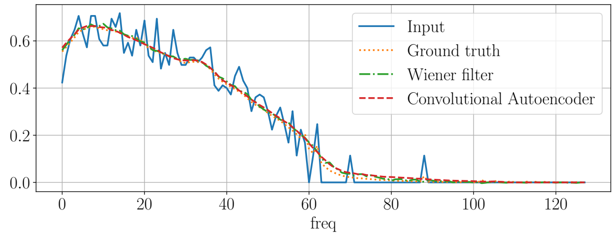


Fig. 7: Comparison of different methods on row 50 of a specific image. The trend on other images is the same.

The operating point is 90keV and 0.2mA. For each acquisition, we place an object in front of the sensor, we irradiate it with an X-ray beam, and record the linear sensor output. For each object we acquire $10\,000$ static images (i.e., the item is steady in front of the sensor) changing the acquisition time $\tau = \{1\text{ms}, 1.5\text{ms}, 2\text{ms}\}$ (i.e., the active recording time of the sensor, similar to a camera shutter time). We divide the frequency spectrum of the acquisition in 128 bins. The final shape of each measure is therefore 128×128, where the first dimension is the number of pixels in a single column, so acquired in the vertical direction, and the second is the number of bins used to represent the spectrum. Different plastic polymers of different thickness are considered as objects, as listed in Table II. These polymers have a chemical composition which represents the most common contaminants that can be found during food inspections. For each material width δ and for each acquisition time τ three measures have been done, totaling 138 measures. The whole corpus of images is therefore composed of $1\,380\,000$ hyperspectral images of shape 128×128.

As both Wiener filtering and autoencoders rely on the offline estimation of a set of parameters (i.e., power spectrums and network weights), we apply a commonly adopted training-validation-testing split policy to avoid bias the achieved results. We pay attention to keep in the same set measurements of the same material (including width δ) with the same acquisition time τ. Hence, we randomly use 70% of the data for training (of which 21.4% for validation) and 30% for testing.

Training and testing pipeline. We develop our autoencoder model using Keras with TensorFlow as backend [14]. The network is trained to minimize the loss function $\mathscr{L}$ computed on batches of $B = 50$ noisy images and their denoised version $\hat{\mathbf{I}}$:

TABLE II: Considered materials for the acquisition

Material	**Label**	**Width δ (mm)**
Background	BKG	-
Polyethylene	PE	2, 8, 16
Polyamide	PA66	2, 8, 16
Polyoxymethylene	POM	2, 8, 16
Polytetrafluoroethylene	PTFE	2, 8, 16
Polyvinyl chloride	PVC	2, 8, 16

$$\mathscr{L} = \frac{1}{B}\sum_{b=1}^{B} \|\mathbf{I}_b - \hat{\mathbf{I}}_b\|_F^2 = \frac{1}{B}\sum_{b=1}^{B} \|\mathbf{I}_b - \mathcal{A}(\boldsymbol{\theta}, \mathbf{I}_b)\|_F^2, \quad (7)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. By minimizing $\mathscr{L}$ we force the denoised image $\hat{\mathbf{I}}$ to be as close as possible to $\mathbf{I}$. We use Stochastic Gradient Descent with Nesterov momentum as optimizer, setting initial learning rate to 0.01, learning rate decay to 10^{-6} each epoch and momentum to 0.9, until reaching convergence on a validation plateau. At testing time, we freeze the network weights and we compute the output from each image in the testing dataset.

V. Results

Table III shows the results for the two considered denoising algorithms in term of Peak Signal-to-Noise Ratio (PSNR), which is defined as:

$$\text{PSNR} = 20 \cdot \log_{10}\left(\frac{HW}{\|\mathbf{I} - \hat{\mathbf{I}}\|_F^2}\right), \quad (8)$$

where $\mathbf{I}$ and $\hat{\mathbf{I}}$ are the compared images of size $H \times W$ whose range is $[0, 1]$, and the numerator term HW is needed to compensate for the signals' size. This particular metric gives us the fidelity of the reconstruction of $\hat{\mathbf{I}}$ w.r.t. $\mathbf{I}$. Even though both algorithms yield good results (approximately 9 dB gain), it is worth noting that the implementation of Wiener filter is heavily tailored to the considered acquisition system data model. Nevertheless, PSNR between the two methods is comparable, showing that it is possible to work in a CNN fashion for hyperspectral Xray denoising. In particular, the advantage of using the autoencoder is that we do not need to have any kind of prior knowledge on the input data model,

TABLE III: PSNR for the considered methods. Best results in bold.

Material **(δ, τ)**	**Noisy acquisition** **PSNR (dB)**	**Wiener filter** **PSNR (dB)**	**Autoencoder** **PSNR (dB)**
BKG 2ms	30.398	**41.192**	38.729
PA66 16mm 1.5ms	31.185	**40.827**	**40.827**
PA66 2mm 1.5ms	30.665	**39.720**	39.406
PE 2mm 1.5ms	30.561	**39.259**	38.852
PE 8mm 1.5ms	30.797	**39.830**	39.721
POM 16mm 2ms	31.148	**41.329**	40.880
POM 2mm 2ms	30.389	**40.479**	39.371
POM 8mm 1.5ms	30.991	**40.362**	40.163
PTFE 16mm 1ms	31.879	**41.696**	40.300
PTFE 2mm 1.5ms	30.752	**39.900**	39.626
PTFE 8mm 2ms	30.835	**40.605**	40.015
PVC 16mm 1ms	32.021	**40.897**	37.289
PVC 16mm 2ms	32.015	**42.649**	40.605
PVC 2mm 2ms	30.513	**40.589**	39.521

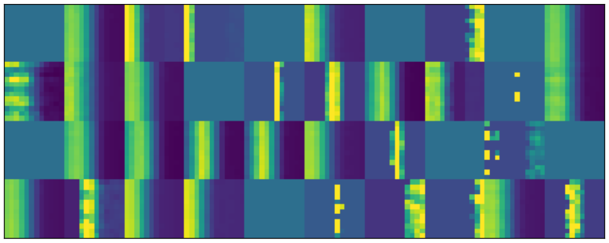


Fig. 8: Hidden layers representation

and we only need some training images. This leads to better generalization capability with respect to Wiener filter method, which requires a carefully characterization of the Noise-to-Signal ratio S_N / S_I.

Moreover, the good denoising results achieved by the Convolutional Autoencoder could be further exploited in case supervised problems (such as food contaminant detection and segmentation) should be faced after denoising. As a matter of fact, a transfer learning approach [15] could be adopted in order to develop and train different CNNs that share many layers with the one proposed in this paper, dramatically reducing the number of parameters to be trained.

A visual example of the goodness of output images with respect to the ground truth is shown in Fig. 6 and a per-pixel plot is depicted in Fig. 7.

An interesting aspect to analyze is the output of the hidden layers of the trained network. In fact, output from hidden encoding layers displays the automatically extracted features from the input image, giving us a clue on what the network is highlighting during the encoding procedure. Fig 8 displays some of the output from the innermost hidden layer, generated by using as input a sample from the measure ”PVC 8mm at 2ms”. We can clearly see different neurons activating at different small frequency bands, each one capturing different shapes and intensities that will be used in the decoder to reconstruct the final output. This enables us to consider the autoencoder as a methodology learning a set of meaningful basis for data projection.

VI. Conclusions

In this paper we investigate denoising strategies for X-ray hyperspectral images. The considered pipeline is composed of a preprocessing step necessary for normalizing input images, followed by candidate denoising solutions. Specifically, we compare the effect of a Wiener filtering approach and a Convolutional Autoencoder. Results show that the solution based on autoencoder provides close results to the one based on Wiener filter. This is interesting as it is a strong indicator that the proposed Autoencoder is capable of learning a meaningful reduced dimensionality representation of the input data. This opens the door to applications that can be built on top of the denoising algorithm (e.g., food contaminant detection, anomaly detection, material recognition, etc.). Indeed, it could be possible to exploit the deep representation of the input data provided by the autoencoder as candidate denoised feature vector to be used for classification purpose.

References

[1] Norbert Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, The MIT Press, 1964.

[2] Vladimir Katkovnik, Alessandro Foi, Karen Egiazarian, and Jaakko Astola, “From local kernel to nonlocal multiple-model image denoising,” *International Journal of Computer Vision*, vol. 86, no. 1, pp. 1–32, jul 2009.

[3] Cristovao Cruz, Rakesh Mehta, Vladimir Katkovnik, and Karen O. Egiazarian, “Single image super-resolution based on wiener filter in similarity domain,” *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1376–1389, mar 2018.

[4] Chunwei Tian, Yong xu, Lunke Fei, and Ke Yan, “Deep learning for image denoising: A survey,” *preprint arXiv:1810.05052,2018*, 10 2018.

[5] Cristovao Cruz, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, “Nonlocality-reinforced convolutional neural networks for image denoising,” *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1216–1220, aug 2018.

[6] Filippos Kokkinos and Stamatios Lefkimmiatis, “Deep image demosaicking using a cascade of convolutional residual denoising networks,” in *Computer Vision – ECCV 2018*, pp. 317–333. Springer International Publishing, 2018.

[7] L. Gondara, “Medical image denoising using convolutional denoising autoencoders,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec 2016, pp. 241–246.

[8] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning - ICML '08*. 2008, ACM Press.

[9] Junyuan Xie, Linli Xu, and Enhong Chen, “Image denoising and inpainting with deep neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 341–349. Curran Associates, Inc., 2012.

[10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[11] David J. Griffiths, *Introduction to Quantum Mechanics*, Prentice Hall, 1994.

[12] Jianming Xu and Buon Nguyen, “CMOS image sensor based x-ray detector noise characterization and its fixed pattern noise correction method,” in *Medical Imaging 2011: Physics of Medical Imaging*, Norbert J. Pelc, Ehsan Samei, and Robert M. Nishikawa, Eds. mar 2011, SPIE.

[13] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[14] François Chollet et al., “Keras,” https://keras.io, 2015.

[15] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu, “A survey on deep transfer learning,” in *ICANN*, 2018.