UNIVERSIDAD DE LA REPÚBLICA URUGUAY

FACULTAD DE INGENIERÍA UDELAR

Statistical learning and convolutional neural networks for supervised and unsupervised restoration of satellite images in low light conditions

Manuel Sánchez Laguardia

Proyecto de grado presentado a la Facultad de Ingeniería de la Universidad de la República en cumplimiento parcial de los requerimientos para la obtención del título de Ingeniero en Sistemas de Comunicación.

Tutores

Charles Kervrann
Emmanuel Moebel
Sébastien Herbreteau

Montevideo, Uruguay
Setiembre de 2023

End-of-studies Internship Report

Statistical learning and convolutional neural networks for supervised and unsupervised restoration of satellite images in low light conditions

Masters thesis for Engineering Diploma in Mathematical and Computational Engineering (MCE)
Master 2 EEA Signal, Image, Embedded Systems and Automation (SISEA)

Manuel SÁNCHEZ LAGUARDIA
manuel.sanchez-laguardia@imt-atlantique.net

Entreprise Supervisors:	Charles Kervrann	Head of SAIRPICO
	Emmanuel Moebel	Postdoctoral researcher
	Sébastien Herbreteau	PhD Student
School Supervisor:	Carlos Granero Belinchón	Professor at IMT Atlantique
Entreprise:	Inria Centre of Rennes University	
Project-Team:	SAIRPICO	
Internship Period:	From 01/04/2023 to 30/09/2023	

Date: August 31, 2023

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to express my sincere thanks to my mentor during this internship, Charles Kervrann, as well as Emmanuel Moebel for trusting me with this internship, for giving me the opportunity to work in such an amazing team and for their constant guidance. I would also like to thank PhD student Sébastien Herbreteau and for his guidance and advice during this past few months.

I would like to extend my warmest gratitude to all my colleagues on the SAIRPICO team, in particular, Yasmine, Sarra, Maelys and Antoine, with whom I shared most of my time at Inria.

Special thanks to Carlos Granero Belinchón, my school supervisor, for his valuable advice, as well as the professors at IMT Atlantique, in particular, Ronan Fablet and Mélanie Blanchard for their help and for the experience they have shared with me over the past two years.

I would also like to thank Claudina Rattaro, Director of my career at Universidad de la República, and Pablo Monzón, coordinator of the exchange with IMT Atlantique. Without them, I would not have had the privilege to be doing my internship at Inria.

Lastly, I would like to thank my family and friends for their unwavering support and for always being there for me.

Summary

This manuscript is the result of Manuel Sánchez Laguardia's end-of-studies internship as part of his engineering studies at IMT Atlantique and Universidad de la República. The development of this work began the 1st of April 2023 and extended until the 30th of September 2023, for a total duration of six months.

The internship was carried under the mentorship of Charles Kervrann and Emmanuel Moebel. It covered two main projects, both related to the subject of this internship: "Statistical learning and convolutional neural networks for supervised and unsupervised restoration of satellite images in low light conditions"

This document is divided into 7 main Sections. First, the introduction to the mission, where the context, background and expected results are explained. Then, the presentation of the hosting organization. Next, the developed work and methodology, where the two main projects are presented in separate Sections. Then, the conclusions followed by the perspectives of the project. Finally, a reflection on my professional project for the future and its link with this internship.

The first project consisted in studying the statistics behind the gradient descent used to estimate parameters of convolutional neural networks. This was achieved via the study of a well-known machine learning algorithm used for denoising: Deep Image Prior [1]. This analysis gave very interesting results. It showed that the direction of the parameters vector of the neural network throughout the iterations, could be explained almost entirely with just one PCA (Principal Component Analysis) vector. However, it also showed that from one iteration to the next, the parameters change almost randomly, and no information could be extracted from them as whole. This first project was set aside to be worked on, possibly, towards the end of the internship.

The second project consisted in performing image restoration methods by combining denoising and deconvolution, using different techniques. I focused on satellite images in low light conditions, as this is what the team has been working since they partnered up with Airbus Space and Defense. The goal was to explore how this different methods performed, and draw conclusions in terms of performances (PSNR) and time computations. It was found that the type of noise had a high impact on the result of the methods. Also, it was shown experimentally that training one of the supervised methods using microscopy images, which are similar to night satellite images, produces very good results and are a good fit for the training phase.

1. Introduction

The main objective of this work was to evaluate, analyse, design and develop both supervised and unsupervised image processing approaches, that exploit convolutional networks and deep learning, applied to night satellite images.

In the first part of the internship, the goal was to study the statistics behind the gradient descent involved in a specific deep learning algorithm dedicated to unsupervised image restoration (Deep Image Prior [1]). The possibility to develop an algorithm from this study was also assessed.

The second part of the internship consisted in evaluating and analysing different denoising and deconvolution algorithms to restore degraded night satellite images.

1.1. Context

The scope of this subject is quite large, as there are many areas of application that could benefit greatly from such work. For instance, one of the partners of the SAIRPICO team is Airbus Space and Defense, require a fast and efficient processing of images captured by satellites. Some other fields in which this work can be applied are, for example: biological and medical imaging, autonomous mobility and transportation, surveillance...

1.2. Challenges for the hosting organization

"With digital technology disrupting all established frameworks, in order to bring through ambitious scientific and entrepreneurial projects, Inria's responsibility is to create value on a large scale for society and for the economy; to raise France's profile and boost its attractiveness; and to increase the impact of research and innovation in the digital sector". The SAIRPICO team within Inria, which is where this internship took place, has some of its own specific challenge which is to investigate new imaging techniques and processing methods, mathematical models, and algorithms to build an integrated imaging system. As for this internship, a particular challenge aligned with all the previous ones consisted in developing a fast and efficient denoising and deconvolution algorithm and proof-of-concepts to be used on images with low-photon counts (microscopy images, night satellite images).

1.3. Expected results

The expected results for the first part were to be able to characterize the statistics of the gradient descent and decipher the behaviour behind deep learning algorithms in convolutional neural networks.

The expected results for the second part were to be able to evaluate and compare different night image processing strategies and algorithms; using various different tools, some of which were developed by the SAIRPICO team in the past years.

2. Presentation of the hosting organization

The internship was carried out at the SAIRPICO (Space-time imaging, artificial intelligence and computing for cellular and chemical biology) team as part of the Inria (National Institute for Research in Digital Science and Technology; French: "Institut national de recherche en sciences et technologies du numérique") Center of Rennes University.

2.1. Inria

Inria is the French national research institute for digital science and technology. It is a Public Scientific and Technical Research Establishment (EPST) under the double supervision of the French Ministry of National Education, Advanced Instruction and Research and the Ministry of Economy, Finance and Industry. It was founded by the French government in 1967 to act as a bridge between the academic world and industry, with their roots in applied mathematics and IT.

Today, Inria continues to support the digital transformation of science, the economy and society as a whole. Inria has 9 research centers distributed across France (in Bordeaux, Grenoble-Inovallée, Lille, Lyon, Nancy, Paris-Rocquencourt, Rennes, Saclay, and Sophia Antipolis) and one center abroad in Santiago de Chile, Chile. It also contributes to academic research teams outside of those centers.

World-class research, technological innovation and entrepreneurial risk are its DNA. In 215 project teams, most of which are shared with major research universities, more than 3,900 researchers and engineers explore new paths, often in an interdisciplinary manner and in collaboration with industrial partners to meet ambitious challenges. As a technological institute, Inria supports the diversity of innovation pathways: from open source software publishing to the creation of technological startups.

2.2. SAIRPICO

The *SAIRPICO Project-Team* [2] is a joint collaboration between the Inria Center of Rennes University [3] and the *INSERM Institut Curie* [4] (U1143 INSERM “Sub-cellular Structure and Cellular Dynamics” Unit). The Project-Team develops its activity primarily in the medical field, with the objective of deciphering the dynamic coordination and organization of molecular complexes at the single cell level. Mathematical theories and algorithms are mainly developed to identify molecular processes in fundamental biology but they have also a strong potential for applications in biotechnology and medicine. The images are often multidimensional and multi modal light microscopy combined with GFP (Green Fluorescence Protein) tagging, but their goal is to investigate new imaging techniques and processing methods, mathematical models, and algorithms to build an integrated imaging approach that bridges the resolution gaps between the molecule and the whole cell, in space and time. This expertise in low-photon count imaging has been translated recently to the aeronautical field after their partnership with the European multinational aerospace corporation: *Airbus Space and Defense*. Several PhD students analyse satellite images and develop methods for noise removal and optical flow computation.

3. Study of the Statistics of Gradient Descent using PCA

3.1. Objective

The main objective was to investigate statistics behind the gradient descent involved in the estimation of neural network parameters. This was achieved by building a bi-modal training algorithm applied to a U-Net neural network. The first mode being the usual gradient descent, and the second mode being the perturbation of the network's parameters, in different ways, and searching for a decrease in the energy (or loss). The focus was on noise removal in images by applying the *Deep Image Prior* [1] algorithm and a deep neural network with hourglass architecture.

3.2. Background

3.2.1. Deep Image Prior

Deep Image Prior (DIP) [1] is an image restoration method that consists in using an hourglass neural network and training with stimulated uniform noise as input and minimizing the distance between the input noisy image and the unknown image to restore. The idea behind this algorithm, is to build a path of gradual minimization of the loss through the unknown noise-free image, before over-fitting to the noisy image.

The goal is to solve the inverse problem of denoising from a corrupted observation y. To achieve this, the neural network is interpreted as a parametrization $x = f_\theta(z)$ of an image $x \in \mathbb{R}^{C\times H\times W}$; where $z \in \mathbb{R}^{C\times H\times W}$ is a code tensor/vector and θ are the network parameters. The denoising problem can be expressed as an energy minimization problem:

$$x = \min_x D(x; y) + \mathcal{R}(x), \tag{1}$$

where $D(x; x_0)$ is a task-dependent data term, y the noisy image, and $\mathcal{R}(x)$ a regularizer. This regularizer is replaced by the implicit prior captured by the neural network. The problem is formulated as follows:

$$\theta^* = \operatorname*{argmin}_\theta D(f_\theta(z); y), \text{ such as } x^* = f_{\theta^*}(z). \tag{2}$$

Given a target image y, the goal is to estimate the parameters θ^* that reproduce y. This can be set up by considering the optimization in (1), and using as data term that compares the estimated image x and the target image y as follows:

$$D(x; y) = ||x - y||^2. \tag{3}$$

Plugging (3) in (1) leads to the optimization problem presented:

$$\min_x ||f_\theta(z) - y||^2 \text{ , with } z \sim U(0, 1) \text{ (Uniform noise).} \tag{4}$$

Once the algorithm is set up and running, the crucial step called Early-Stopping (ES) consists in stopping the training procedure when the output, which is unknown at time t, is very close to the noise-free image, before it over-fits. This idea is illustrated in Figure 1. It is expected that the generated image will be, at some *Optimal Stopping point* labelled in red in Figure 1, close to the noise-free image. Continuing iterating the algorithm generates an image close to the initial input degraded image y. Different ES methods (e.g. Total Variation, Variance) are analysed and tested in the following sections.

This algorithm, in particular, was selected as subject of study because it has the advantage of analysing only one image at a time, and having an extensive learning phase from which to draw conclusions about the gradient descent. Consequently, one epoch corresponds to one iteration in the DIP framework.

3.2.2. Total Variation (TV)

TV identifies the variation in the structure of the image, and can be defined as the absolute value of the gradient of the value of each pixel in the image.

$$TV(x) = \int_\Omega |\nabla_x(p)| dp, \tag{5}$$

where $\nabla.$ denotes the gradient operator, Ω is the image domain and $p \in \Omega$ designates a pixel location in Ω.

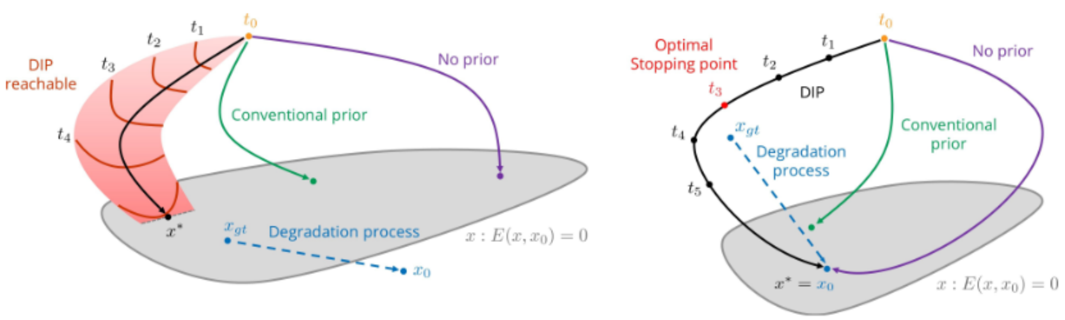


Figure 1: Deep Image Prior explanation

3.2.3. Mean and Variance

The sample mean μ and sample variance σ^2, are two statistical moments that give very relevant information about the distribution of a sampled random variable. Given $\theta \in \mathcal{R}^n$, $\theta = (\theta_1, \theta_2, ..., \theta_n)$, with n the number of parameters and N the number of samples:

$$\mu = \frac{1}{n} \sum_{i}^{n} \theta_i, \tag{6}$$

$$\sigma^2 = \frac{1}{n} \sum_{i}^{n} (\theta_i - \mu)^2. \tag{7}$$

3.2.4. Metropolis-Hastings Algorithm

The Metropolis-Hastings (MH) algorithm is the most popular and widely applied MCMC algorithm, and it is based on the MH procedure [5]. This algorithm consists in accepting or rejecting the new parameters vector, with a certain probability, even when the energy increases from one iteration to the next. This would allow the optimization process to move from a possible local minimum and search for a path leading to the global minimum.

The following explanation was inspired from [6]. This algorithm involves the definition of a proposal density $q(\mathrm{z}|\mathrm{x}), x, z \in \mathcal{X}$ to move from state x to state z, and the acceptance probability $0 \leq a(x, z) \leq 1$. Then, the transition probability is defined as $p(z|x) = q(z|x)a(z, x)$. A sample z is drawn from the proposal distribution and then a test is applied to accept the transition from state x to z or not. In this case, the sample z consists in taking the mean vector and adding the variance multiplied by a factor γ:

$$z = \mu + \gamma\sigma^2. \tag{8}$$

In practice, this reduces the sampling procedure to drawing a sample γ from a uniform distribution $U(-1, 1)$, as follows:

1. Set an initial state x_0.
2. For $t = 1, \dots, T$ do
   (a) Draw a sample $\gamma \sim U(-1, 1)$ and compute $z = \mu + \gamma\sigma^2$.
   (b) Compute the acceptance probability: $a\left(x^{(t-1)}, z\right) = min\left[1, \frac{\pi(z)q(x^{(t-1)}|z)}{\pi(x^{(t-1)}q(z|x^{(t-1)})}\right]$.
   (c) Draw α from a uniform distribution: $\alpha \sim U(0, 1)$.
   (d) If $\alpha \leq a(x^{(t-1)}, z)$ then $x^{(t)} = z$, else $x^{(t)} = x^{(t-1)}$.

3.2.5. Principal Component Analysis

The Principal Component Analysis (PCA) is used to study the behaviour of a set of N-sample of random vectors. The PCA procedure is based on the analysis of the correlation matrix $\hat{\Sigma}_X$ empirically defined as:

$$\hat{\Sigma}_X = \frac{1}{N} D^T D, \text{ where } D = (X_1, ..., X_N) \text{ and } X_k = (x_1, ..., x_n)^T. \tag{9}$$

In our context, the vectors X_k correspond to the parameters of the neural network θ at each iteration k within the established interval, whereas the x_j correspond to the actual values of the parameters at position j in the vector. In general, n is very large as the number of parameters is around 5×10^5 to 2×10^6 (depending on the network's architecture).

In this case, as the number of samples N is much smaller than n (about 10^3 to 5×10^3), this approach would imply calculating a $n \times n$ matrix, which would be very computationally expensive. To avoid this, it is possible to calculate the $N \times N$ matrix T defined as:

$$T = \frac{1}{N} D^T D. \tag{10}$$

Then, the $N - 1$ eigenvalues (κ_i) and eigenvectors (e_i) of T are:

$$T e_i = \kappa_i e_i, \quad i = 1, ..., N - 1. \tag{11}$$

By multiplying the eigenvectors by D on both sides, and replacing T with its formulation:

$$\begin{gathered} \underbrace{\frac{1}{N} D D^T}_{} D e_i = \kappa_i D e_i, \\ \hat{\Sigma}_X \left(D e_i \right) = \kappa_i \left(D e_i \right). \end{gathered} \tag{12}$$

If e_i is an eigenvector of T, then De_i is an eigenvector of $\hat{\Sigma}_X$, and its eigenvalues (λ_i) verify that: $\lambda_i = \kappa_i$. The eigenvectors of $\hat{\Sigma}_X$ are defined as:

$$\phi_i = \frac{1}{\sqrt{\kappa_i N}} D e_i. \tag{13}$$

With this eigenvalues and eigenvectors, it is possible to create new samples $\tilde{X}$ for the parameters vector θ and load it into the network, in order to observe the output. This was achieved by taking the mean ($\bar{X}$) of the parameters, at the interval of interest, and adding or subtracting the most relevant eigenvectors multiplied by a certain factor of magnitude (α):

$$\tilde{X} = \bar{X} \pm \alpha \sqrt{\lambda_i} \phi_i. \tag{14}$$

3.3. Experiments

3.3.1. Peak-Signal-to-Noise Ratio (PSNR)

The PSNR value measures the ratio between the maximum possible power of a signal and the power of corrupting noise. PSNR is usually expressed as a logarithmic quantity using the decibel scale. PSNR is most easily defined via the mean squared error (MSE). Given a noise-free $H \times W$ monochrome image X and its noisy approximation Y, MSE is defined as:

$$MSE = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (X(i,j) - Y(i,j))^2. \tag{15}$$

Then the PSNR (in dB) is defined as follows:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_X^2}{MSE} \right), \tag{16}$$

where MAX_X corresponds to the maximum pixel value of image X.

3.3.2. Total Variation

The preliminary tests carried out, consisted in identifying possible ways to early-stop the Deep Image Prior (DIP) algorithm, inspired from previous papers [7].

As a starting point, the variance at each iteration was calculated and DIP is early-stopped when it is at its lowest value. The Total Variation (TV) of the image was another possible candidate criterion for Early-Stopping DIP. The idea is to detect a specific behaviour of this function. The TV is assumed to be high during the first iterations and decrease as noise is continuously removed. The mentioned TV is expected to coincide with the ground truth image as soon as it is close to the ground truth. At the end of the learning procedure, TV increases as the algorithm over-fits to the noise.

In our experiments, it was found that this was not the case. The first outputs of the network are very blurry images, which had almost no high gradient values, and a very small TV value observed. In Figure 2, the blue line shows the TV value of the model's output at each iteration, the red dotted line showing the TV value of the ground truth image, and the yellow dotted line corresponding to the iteration closest to the ground truth. The model outputs for the first 1000 iterations are shown in Figure 3. It can be observed that the TV value is actually small at the beginning because the output image is very blurry and has little to no contrasting areas. The main cause of this result is the network's architecture, as it has a down-scaling phase that reduces the resolution of the image before enlarging it again at the end. As the input is just noise, and the network has no trained relevant weights, then this resolution modification process plus the convolutional layers give this blurry noise as outputs, which have a very low TV value.

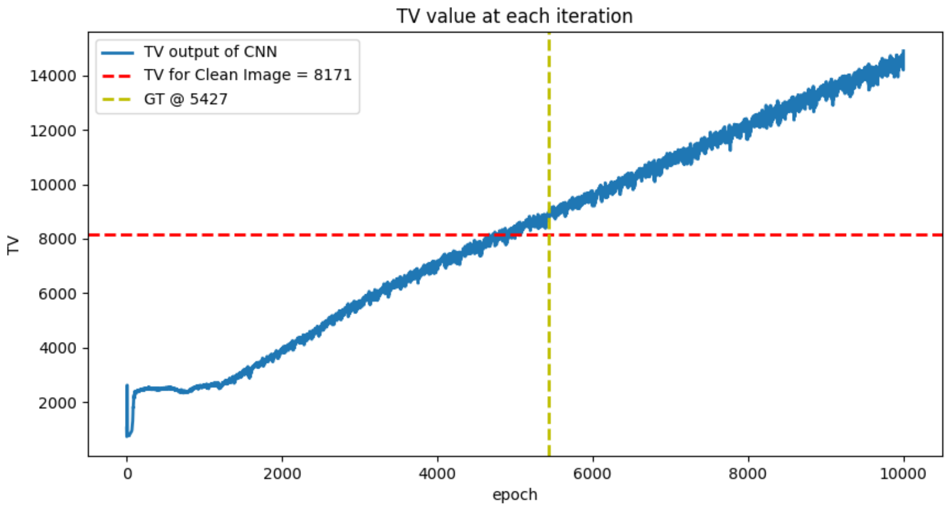


Figure 2: TV values at each iteration of Deep-Image-Prior.

3.3.3. Mean, Variance and Metropolis-Hastings procedure

The following experiment consisted in capturing the vector of parameters for a couple of iterations and calculating its sample mean and sample variance, respectively.

After setting the mean vector μ as the network parameters, the idea was to perturb a given vector with its standard deviation σ and calculating the loss, hoping for a decrease in energy or loss. It was necessary to separate the "warm-up" phase, where the output image was just a blurred noise, from a second phase where the contents of the input image started to be visible. The parameters were always captured from the second phase, as the first one is not as relevant.

In the first experiments, the Metropolis-Hastings algorithm was used to minimize the energy $||f_\theta(z)-y||^2$. The DIP method was run until the "warm-up" method is finished, and a significant amount of parameters vectors is available. The mean and variance of this vectors are computed and the MH algorithm is started. The experiment showed that the energy stayed almost constant after changing modes, as illustrated in Figure

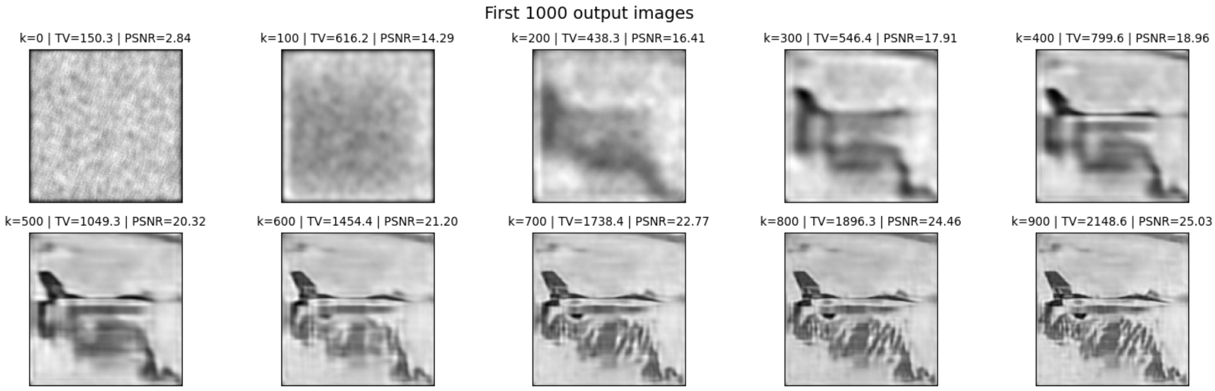


Figure 3: Output images after $k < 1000$ iterations of DIP. The PSNR and TV values are given at each iteration.

4, or skyrocketed to very high values, but never decreased considerably. The next attempt was to modify the MH algorithm's acceptance condition to only accept the values that decrease the energy, but this did not work either because of the high dimensionality of vectors.

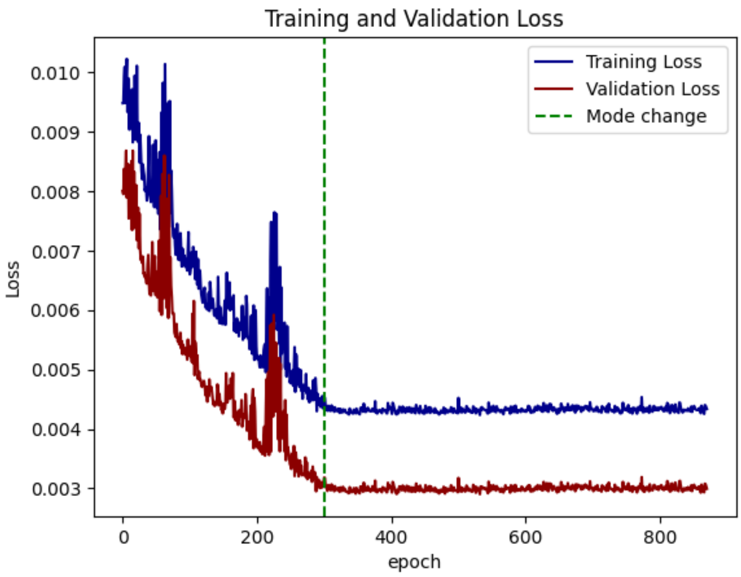


Figure 4: Training and Validation Loss for two modal training using Metropolis-Hastings algorithm

3.3.4. Principal Component Analysis

The next experiment consisted in performing a Principal Component Analysis (PCA) of the parameters vector in a set interval of iterations. Different interval sizes were tested, as well as different images.

It was observed that, in general, most of the new orthogonal basis, resulting from the PCA, could be explained by just 1 eigenvector ϕ_1. Its eigenvalue λ_1 contributed with more than 90% of the total weight, whereas the other $N-2$ eigenvalues contributed with almost nothing. This means that generating new samples is possible just by varying the first eigenvector (ϕ_1), which is very computationally convenient.

Secondly, a similar analysis was performed on the difference between two consecutive parameters vectors $\Delta\theta$ to studying the variation of the gradient. The squared norm of the difference between one iteration and the previous one was calculated for each iteration and PCA analysis was performed:

$$||\Delta\theta_{\mathbf{t}}||_2^2 = ||\theta_{\mathbf{t}} - \theta_{\mathbf{t-1}}||_2^2. \tag{17}$$

The resulting eigenvalues and vectors showed that the weights were much more spread across all the eigenvalues, in contrast with the PCA applied to sample θ. The importance of the first 20 eigenvalues w.r.t. to the sum of all eigenvalues are reported in Table 1. A priori, this result would imply that it is not possible to extract valuable information from the changes between one iteration to the next, meaning the variations are noise vectors. We computed the empirical distribution of $||\Delta\theta||_2^2$ to verify if it followed a χ^2 law, if the vectors $\Delta\theta$ are Gaussian samples. In a first step, the variation had to be centered and normalized as follows:

$$\tilde{\Delta\theta_i} = \frac{\Delta\theta - \mu}{\sigma}, \tag{18}$$

where μ and σ are the sample mean and sample variance, respectively. Then, the following norm is computed:

$$||\tilde{\Delta\theta_i}||^2 = \sum_{i=1}^{n} |\tilde{\Delta\theta_i}|^2, \tag{19}$$

where n is dimension of the vector. If the vectors are Gaussian distributed, we have:

$$||\tilde{\Delta\theta_i}||^2 \sim \chi^2(n, 2n). \tag{20}$$

A simulation of Gaussian samples was performed, with the same vector size and same number of samples to see what it would look like. This can be observed in Figure 5a (left). However, when the test was performed on the real values, the obtained distribution was not the expected one, suggesting that the vectors are not purely Gaussian distributed (Figure 5b (right)), as some information can still be extracted from this result. The obtained distribution is apparently Laplace distributed, as the main mode is not centered in 0 and it could be approximated as:

$$\text{Laplace}(\mu, b), \text{ with pdf: } f(x|\mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}} \tag{21}$$

Nevertheless, a deeper evaluation exceeds the scope and time-frame of this internship, so it will be left for future work. Instead, the PCA tests were exclusively performed on the parameters vectors.

Table 1: Initial contributions of eigenvalues λ_i for $||\Delta\theta||_2^2$.

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}
3.34 %	2.36 %	2.25 %	2.16 %	1.93 %	1.88 %	1.83 %	1.79 %	1.78 %	1.67 %
λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}	λ_{17}	λ_{18}	λ_{19}	λ_{20}
1.63 %	1.60 %	1.56 %	1.51 %	1.49 %	1.49 %	1.44 %	1.42 %	1.39 %	1.36 %

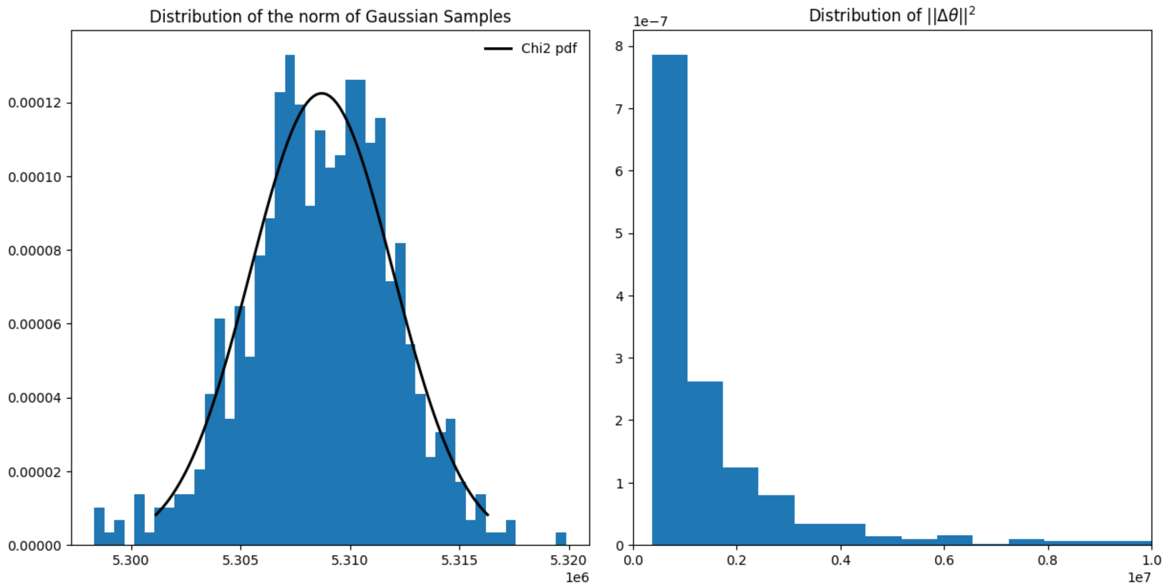
3.3.4.1. Algorithm parameters used in the PCA analysis

All the following tests were performed for various images with added Gaussian noise of $\sigma = 25/255$. The presented graphs below were performed for the "hill" image, shown in Figure 6, unless otherwise specified. The size of the image varied between 256×256 and 512×512 pixels depending on the complexity of the task.

Due to the fact that the noise was added on top of an image without noise, it is possible to calculate a distance between the output of the network and the noise-free image called "ground truth" (GT). This information was exploited in various tests, knowing, however, that it would not be available when applying the experiments to a real case scenario. In a real-world problem, only a noisy image would be accessible, that is when there is no ground truth.

As starting point, the aforementioned PCA was applied to 4 different intervals that are not far from the ground truth:

1. $\Delta_0 = [\text{GT} - 100, \;\; \text{GT}]$



(a) χ^2 distribution from simulated Gaussian samples. (b) Observed distribution of $||\Delta\theta||^2$

Figure 5: Comparison between expected (left) and observed (right) distribution of $||\Delta\theta||^2$.

2. $\Delta_1 = [\text{GT} - 200, \;\; \text{GT}]$
3. $\Delta_2 = [\text{GT} - 500, \;\; \text{GT}]$
4. $\Delta_3 = [\text{GT} - 1000, \text{GT}]$

The importance of the three first eigenvalues of each interval are reported in Table 2. It can be seen that the larger the interval is, the more important the first eigenvalue is with respect to the others.

Table 2: Proportional weights of λ_i for the PCA of θ_i for different intervals.

Interval	λ_1	λ_2	λ_3
Δ_0	79.96 %	6.99 %	3.70 %
Δ_1	86.28 %	4.66 %	2.32 %
Δ_2	94.80 %	1.75 %	0.81 %
Δ_3	94.01 %	2.89 %	1.41 %

3.3.4.2. Analysis of the 1st eigenvector

As the importance of the first eigenvalue λ_1 is much higher than the other eigenvalues, only the first eigenvalue was used to perform the analysis. For each interval Δ_i, the mean squared error between the output and the noisy image, referred to as "Energy", as well as the PSNR between the output and the ground truth (GT), were computed. To be able to perform this comparison, a verification was carried out beforehand, to guarantee that the aggregated intensity of the pixels' value is the same throughout all outputs. The results are illustrated in Figure 7, as well as the output images for the two first intervals Δ_0 and Δ_1 in Figure 8. The images in the middle correspond always to the mean of the entire interval, and moving to the sides, the modified output using the 1st eigenvector. As it can be seen, the addition of the eigenvector can give better results in terms of PSNR with respect to the mean, but in a small amount: between almost 0dB and 0.1dB, depending on the interval and the image. However, taking the mean of the interval, when the interval is small and near the GT is already an improvement of 0.15dB or 0.2dB in the best cases.

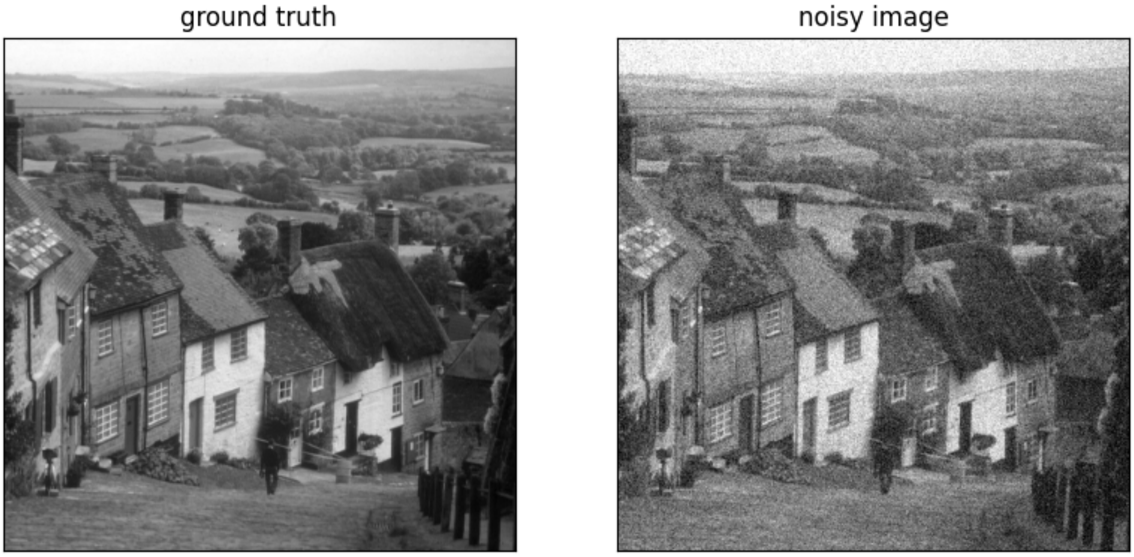


Figure 6: Ground Truth and noisy image

Importance of 1st eigenvalue for 100-iteration windows

The importance of the first eigenvalue, w.r.t. the total sum of eigenvalues, in 100-iteration windows, is analysed for 4 different images. In Figure 9, the graph shows the percentage (between 0 and 1) of the first eigenvalue over the total for various intervals. All intervals were taken after the 500th iteration, in order to be after the "warm-up" period. The first interval Δ_0 goes from 500 two 600 iterations, and is represented at $x = 500$ in the graph. The second interval Δ_2, between 600 and 700, is represented at $x = 600$ in the graph. The third, Δ_3, between 700 and 800, represented at $x = 700$; and so on. It is observed that at the beginning, the 1st eigenvalue represents around 93% of the total eigenvalues, but as the 100-iteration window slides forward, this importance descends until an almost constant value of around 78% after the 4000th iteration. It can also be observed that around the GT of every image, the weight of λ_1 is between 80% and 85%.

3.3.4.3. Analysis of the two first eigenvectors

The influence of adding a second eigenvector ϕ_2 to the equation was evaluated. The results of the varying both ϕ_1 and ϕ_2 at the same time are illustrated in Figure 10a (left). Four three-dimensional graphs are plotted, each one showing on the x-axis the variation of ϕ_1 named α, on the y-axis the variation of ϕ_2 named β, and on the z-axis the variation of PSNR between the output and the ground truth image. Each graph has different interval sizes, which correspond to the ones mentioned before. It can be stated that for most intervals, the highest PSNR value is obtained near the center, meaning that the best image is actually obtained by taking the mean of the interval values without the addition (or subtraction) of eigenvectors. This same analysis was also performed for the same-sized intervals but between the GT and the next 100, 200, 500 and 1000 iterations, instead of the previous ones. The results were very similar.

The equivalent tests were performed to evaluate the variation of energy as a function of ϕ_1 and ϕ_2. The result is shown in Figure 10b (right). As in the PSNR case, for most intervals, the lowest energy value is also obtained near the center, meaning that the closest image to the noisy one is obtained by taking the mean of the interval values.

3.3.4.4. Noise power variation validation

It would also be of interest to verify if PCA could be helpful to generate an image with a smaller noise power, and to see if the process was similar for different noise magnitudes. An image with noise $\sigma_N = 10/255$ was used, as well as the eigenvectors from previous experiments. The energy was calculated

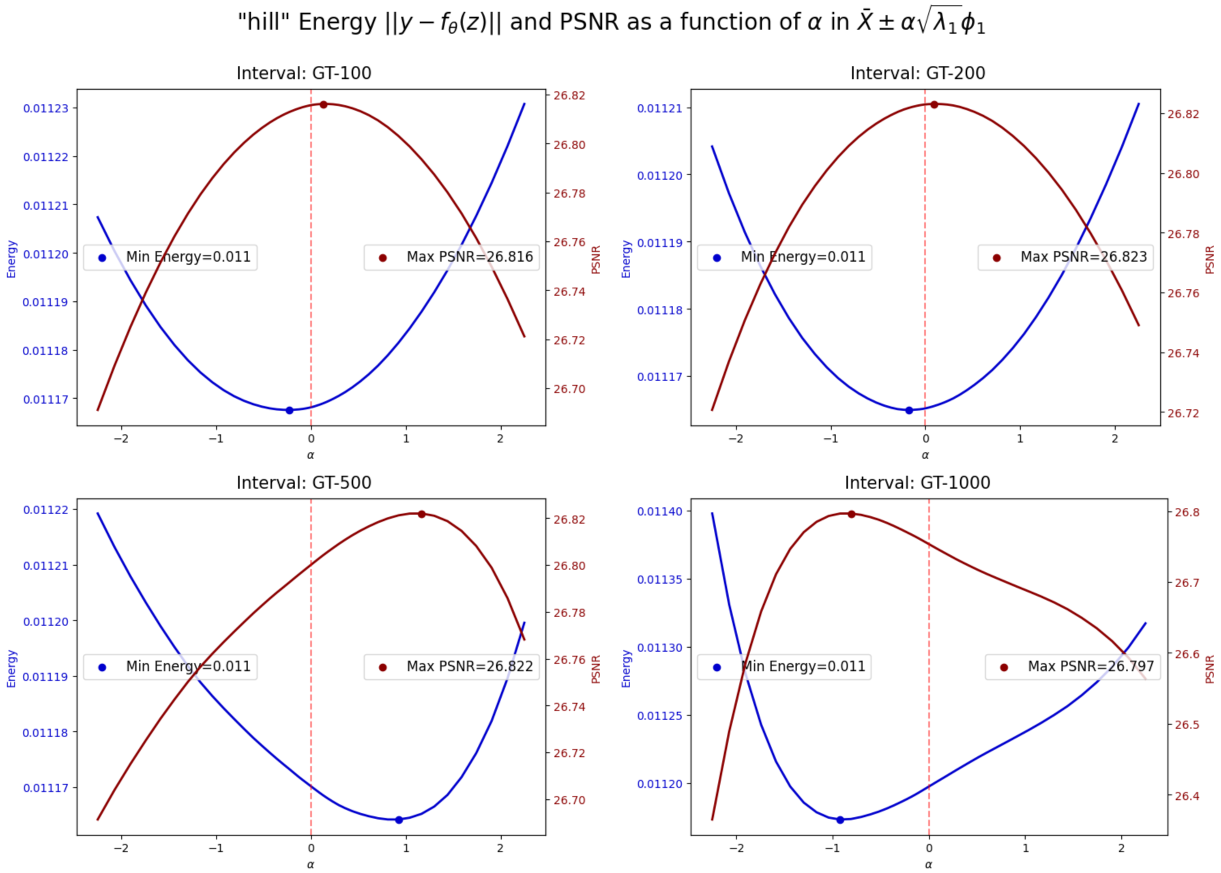


Figure 7: Energy and PSNR values when modifying the output using one eigenvector of the PCA.

and we found that the PSNR had almost the same value as before: around $26.8dB$. Meanwhile, for the classic approach of DIP with an image with noise $\sigma_N = 10/255$, the value of the PSNR is of around $30dB$.

3.3.4.5. Fixed interval sets

In our last experiments, a different set of intervals was chosen. The algorithm run for 10 thousand iterations and the saved parameters' vectors were divided in small batches of 1000 iterations each. At each batch, the PCA of those vectors was performed. The influence of using the first eigenvector was evaluated in terms of the maximum achievable PSNR and the lowest achievable energy. The idea was to determine if the energy would hit a lowest within the interval around the GT ($4000 - 5000$), or if there was a specific behaviour in this interval that was not present in the others. Unfortunately, this was not the case: the lowest energy appeared at the last intervals, when the model is over-fitting, and no particular behaviour was noted in the given interval. The results of the energy are visualized in blue in Figure 12. Alongside the energy, the PSNR to the GT was also calculated and, as expected, the maximum value is in the interval around $4000 - 5000$ iterations.

3.3.4.6. Multiple image sets

The possibility to go from one image to another by performing the PCA on one of the images was studied. This consisted in extracting the eigenvalues of image 1 and minimizing the energy between the modification of the target image, by adding or subtracting the principal eigenvector, and a different noisy image 2.

The resulting image of this test was just noise. This proved that the PCA of the network's parameters is strongly correlated to the *content* of the image and not so much to the statistical properties of the gradient

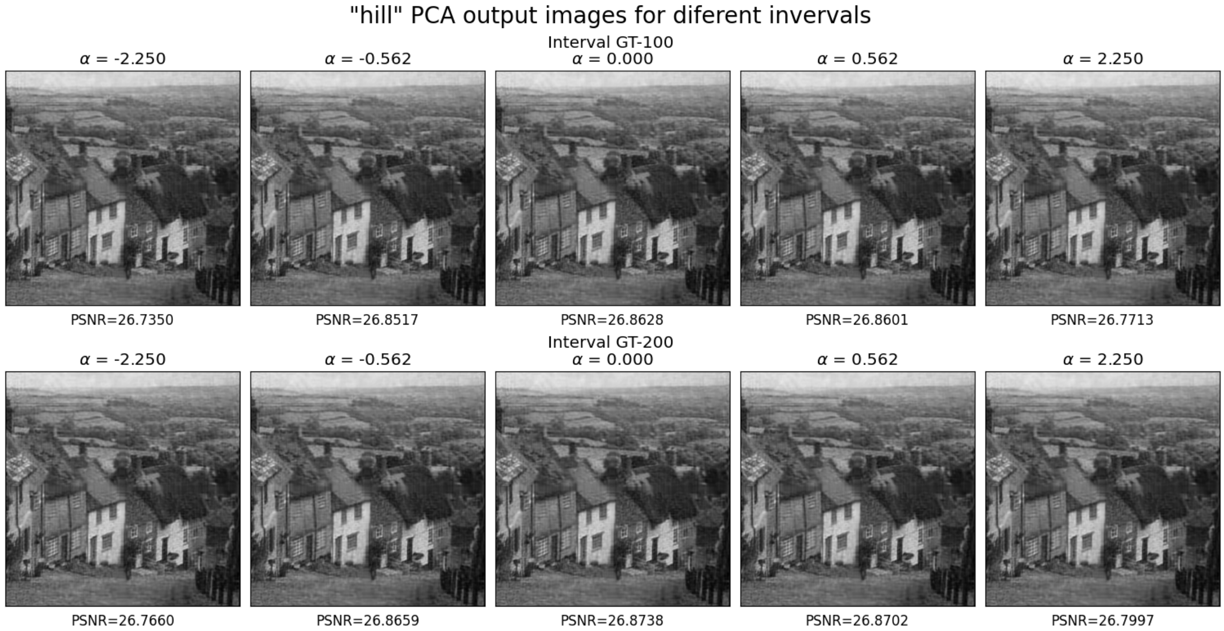


Figure 8: Outputs of the PCA modification of the output using one eigenvector.

descent.

This property holds true as confirmed by another experiment, where a set of M images were passed through the model, 100 samples of parameters for each image were extracted, and the PCA of the $M \times 100$ matrix of all parameters together was performed. It was found that the importance of the resulting PCA eigenvalues was spread out in $M - 1$ eigenvalues, meaning that with two images used, only one eigenvalue had around 99%. With 3 images used, the result was 2 eigenvalues with around 50% of the weight each. With 5 images, the result was 4 eigenvalues with around 25% each, and so on.

Finally, an attempt was made to recover one of the images inside the set of images used, just by varying one eigenvector. It was found that once there were more than 5 images used, it was not possible to recover the image. However, if all eigenvectors were taken into consideration, the image would be able to be recovered, but this would imply such a higher computational cost that it would not make sense. It would be necessary to do the same analysis done in Figure 10 but for as many eigenvectors as number of images in the set (minus one), which would be too complicated. The result of this multiple-image test is illustrated in Figure 13. It shows the recovery of an image from a set of $M = 2, \ldots, 5$ images using PCA and only the first eigenvector ϕ_1.

3.4. Results Summary

For all the experiments described earlier, the main conclusions are the following ones:

- The PCA of the parameters vector from N samples, yields a single very important eigenvector in the orthogonal basis, and almost irrelevant $N - 2$ others.
- The PCA of the difference of the parameters vector between two consecutive iterations shows an opposite behaviour from the simple PCA of the parameters vector. It shows that the variations from one iteration to the next cannot be compressed by using the PCA, but these variations do not correspond to independent Gaussian noise samples. A further in-depth study of its behaviour is required.
- The parameters of the U-Net architecture for the Deep Image Prior are very dependent on the target image and not on the statistical aspect of the gradient descent.

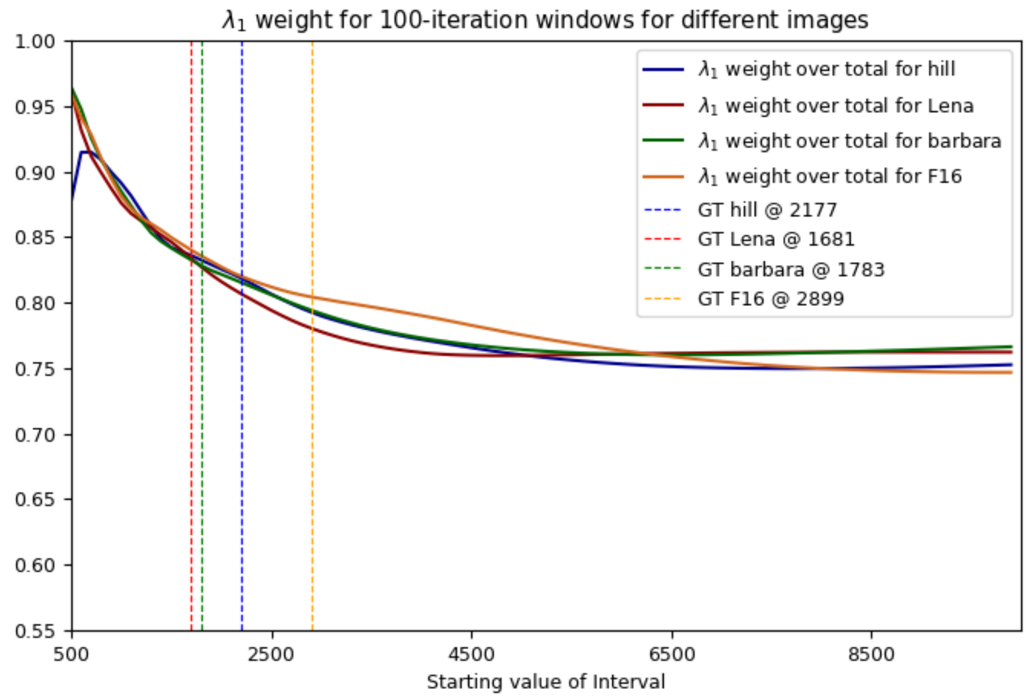


Figure 9: λ_1 weight for different images at different intervals.

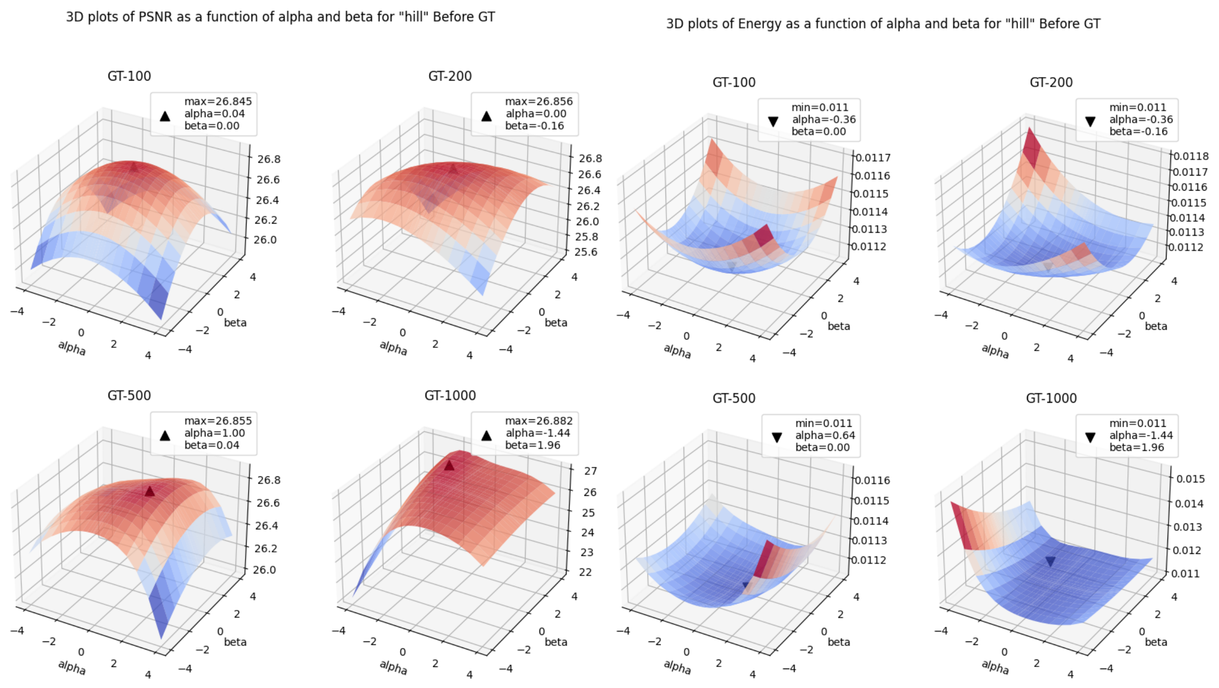
- The resulting image from performing the PCA as presented above is worse, in terms of PSNR, than the output from DIP at the optimal stopping point.

3.4.1. Perspectives

It might be interesting to see if the results of the PCA are suffering from the curse of dimensionality when working with such large vectors θ, as they happen to be in a high-dimensional space.

The entire PCA was performed, in this case, on Deep Image Prior, but could perfectly be applied to other models such as Diffusion Models [16][17][18]. Such an analysis would be out of the scope of this internship but could be subject of further investigation, applying more complex techniques.

It is also worth mentioning that the Deep Image Prior procedure works as a denoising technique mainly because the neural network used is a U-Net, which has a "downsampling" phase, followed by an "upsampling" one. This consists in having the resolution if the image reduced at first and augmented in the end, with other layers, such as convolution layers, in between. This is not explained in the paper [1], and could be subject of further exploration, in order to understand why it works on some networks (e.g. U-Nets) but not other architectures.



(a) 3D PSNR graph as a function of ϕ_1 and ϕ_2 (b) 3D Energy graph as a function of ϕ_1 and ϕ_2

Figure 10: PSNR and energy analysis by adding and subtracting the two first eigenvectors ϕ_1 and ϕ_2 to the mean vector of θ for the 4 different intervals Δ_i.

4. Restoration of night satellite images with supervised and unsupervised methods

4.1. Objective

As part of the partnership between Inria and Airbus, the future deployment of a satellite by Airbus requires the development of innovative fast ways to restore images, in particular, images in low-light conditions. The objective of this part of the internship is to evaluate different denoising and deconvolution techniques, and compare them with state-of-the-art methods.

4.2. Background

4.2.1. Image Restoration

This technique consists in recovering a image that has been degraded in some way. The degradation studied in this work come from different sources. First, noise that appears in the images due to electronic noise in the camera system and can be modeled by Gaussian distribution. This degradation can be written mathematically as in Equation (22), where $\mathbf{y}$ is the observed image, $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ the Gaussian noise, and x the image to recover.

$$\begin{gathered}\text{Noisy Image :}\\ y = x + \varepsilon.\end{gathered} \tag{22}$$

The second type of degradation studied corresponds to the convolution of the image with a certain point spread function (PSF). This function can be expressed as a convolution filter, visually adding blur to the image. This degradation can be written mathematically, where h is a convolution filter and ε the noise:

$$\begin{gathered}\text{Noisy Blurred Image :}\\ y = h * x + \varepsilon.\end{gathered} \tag{23}$$

Figure 11: Comparison between the input image, the output of Deep Image Prior, and the output of the best PCA image obtained. PSNR values are given for each image.

Image restoration is a particularly difficult task because once an image is degraded, information is lost, and its recovery implies making various assumptions of the image properties. For example, the assumption that neighbouring pixels are similar, or the assumption of self-similarity between different areas of the image, are some of the most common assumptions made for natural looking images.

4.2.1.1. Denoising and Deconvolution

As explained above, there are many different types of degradation that may occur to an image. In this case, we focus on the denoising and deconvolution techniques. The denoising method, as its name suggests, implies the removal of noise present in the image. On the other hand, deconvolution consists in the recovering of an underlying image that has been blurred by a known point spread function h, by performing the operation inverse to the convolution. This technique requires the knowledge, or estimation, of h.

4.2.2. The Richardson-Lucy approach

Even though it is more than 50 years old, Richardson-Lucy (RL) deconvolution [10] [11] is still widely used, in multiple fields, because of the good results it achieves. In particular, it has a direct application in astronomy, since it is a deconvolution technique that is mainly used to tackle images corrupted by Poisson noise. RL is a Bayesian-based iterative method that consists in iterating Equation 24, which is positive at each iteration:

$$\hat{x}^{(t+1)} = \hat{x}^{(t)} \cdot \left(H^* \frac{y}{H\hat{x}^{(t)}} \right), \tag{24}$$

where H is the matrix representation of the PSF h. A stopping condition is also set to avoid the appearance of "night sky" artifacts [12].

4.2.3. The "Plug and Play" approach

The "Plug and Play" (PnP) method [13] is a powerful optimization technique to solve inverse problems. It relies on proximal algorithms such as Half Quadratic Splitting (HQS) [14], where the regularization term is usually replaced by a denoiser [15]. Such problems require the use of prior knowledge on images to achieve the most likely solution. Traditionally, convex regularization functions are used as priors, such as Total Variation. In this case, the regularization term $\mathcal{R}(x)$ of the optimization equation has been replaced by a denoiser, as demonstrated below.

Let us consider the model (23) and define the Maximum a posteriori (MAP) estimate as follows:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \, ||Hx - y||_2^2 + 2\sigma^2 \mathcal{R}(x). \tag{25}$$

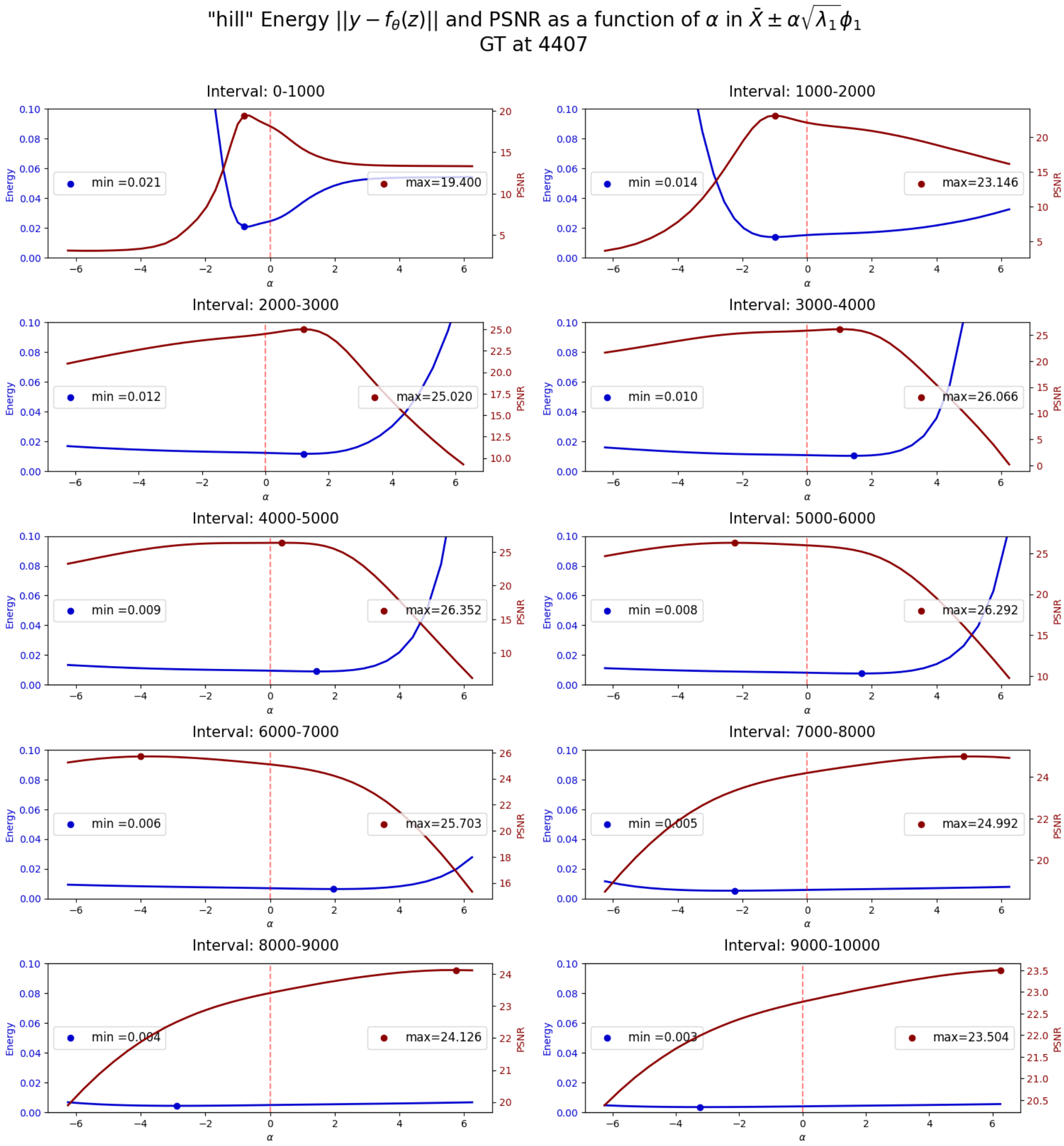


Figure 12: Energy and PSNR as a function of ϕ_1

In the half quadratic splitting framework, an auxiliary variable z is introduced and (25) is relaxed as follows:

$$\hat{x}, \hat{z} = \underset{x,z}{\operatorname{argmin}} \, ||Hx - y||_2^2 + 2\sigma^2 \mathcal{R}(z) \quad \text{subject to} \quad z = x. \tag{26}$$

Reformulating this problem as a penalty, $\exists$[1] $\lambda > 0$, we have:

$$\hat{x}, \hat{z} = \underset{x,z}{\operatorname{argmin}} \, ||Hx - y||_2^2 + 2\sigma^2 \mathcal{R}(z) + \lambda ||x - z||_2^2 \tag{27}$$

Such a problem can be resolved iteratively by "ping-pong"[2]:

$$\begin{cases} x_k = \operatorname{argmin}_x \, ||Hx - y||_2^2 + 2\sigma^2 \mathcal{R}(z_{k-1}) + \lambda ||x - z_{k-1}||_2^2 \\ z_k = \operatorname{argmin}_z \, ||Hx_k - y||_2^2 + 2\sigma^2 \mathcal{R}(z) + \lambda ||x_k - z||_2^2. \end{cases} \tag{28}$$

[1] $\lambda = +\infty$ is particularly suitable, but also any sufficiently large λ.
[2] This guarantees that $q(x_k, z_k) \geq q(x_{k+1}, z_k) \geq q(x_{k+1}, z_{k+1})$.

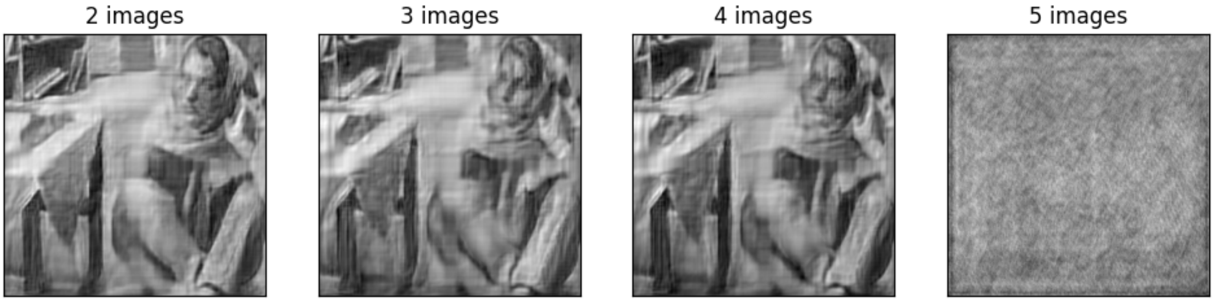


Figure 13: Outputs of the PCA model for multiple images together.

Furthermore:

$$\begin{cases} x_k = \text{argmin}_x \, ||Hx - y||_2^2 + \lambda ||x - z_{k-1}||_2^2, \\ z_k = \text{argmin}_z \, ||Hx_k - y||_2^2 + 2\left(\frac{\sigma}{\lambda}\right)^2 \mathcal{R}(z). \end{cases} \tag{29}$$

It is worth noting that the estimation of z is nothing else than a denoising step. Consequently, both problems can be easily solved as follows:

$$\begin{cases} x_k = \left(H^T H + \lambda I_n\right)^{-1} \left(H^T y + \lambda z_{k-1}\right), \\ z_k = \text{Denoiser}\left(x_k, \frac{\sigma}{\sqrt{\lambda}}\right), \end{cases} \tag{30}$$

where H is the matrix representation of the PSF h. Initially, z_0 is usually set to the observed image y. This two-step restoration method requires the appropriate choice of the parameter λ, which may be a hard task, as explained below.

4.2.3.1. Fast Fourier Transform to speed up calculations

In order to make the Plug and Play algorithm run faster, the Fast Fourier Transform (FFT) was used. In the following, when mentioning convolution, what would actually be more accurate is a cross-correlation, but as "Pytorch" uses the term convolution, it is what will be used for simplicity.

If H is a circular convolution matrix (matrix applying a convolution with circular padding); H^T corresponds to the application of a circular convolution whose kernel is inverted. In particular, $H^T = H$ for a symmetric kernel. Given z_{k-1}, let x_k be such that:

$$(H^T H + \lambda I_n) x_k = (H^T y + \lambda z_{k-1}). \tag{31}$$

It can be seen that applying $(H^T H + \lambda I_n)$ is equivalent to applying a circular convolution matrix, whose kernel is $[K \circ K_{inv} + \lambda I]$. It can be established that the kernel $[K \circ K_{inv} + \lambda I]$ is symmetric. Then, it follows that:

$$F([K \circ K_{inv} + \lambda I]) \odot F(x_k) = F((H^T y + \lambda z_{k-1})), \tag{32}$$

from where the following equation is obtained:

$$x_k = F^{-1}\left(\frac{F((H^T y + \lambda z_{k-1}))}{F([K \circ K_{inv} + \lambda I])}\right). \tag{33}$$

If H was not a circular matrix but a convolution matrix "same" for instance, a circular convolution matrix would have to be created by applying a Hamming window before, so as to blend in the borders of the image and turn the image circular.

This use of FFT enables a faster computation, as there is no need to compute the inverse of the matrix H to achieve the results.

4.2.4. Presentation of image denoisers

In order to perform image restoration with the PnP approach, various denoisers may be considered. Here, we examined the recent NL-Ridge [9] and DCT2net [8] methods. The NL-Ridge denoiser is a Non-Local unsupervised denoising method developed by the SAIRPICO team in 2022. It consists in dividing the target image in patches and denoising each patch by linearly combining the most similar noisy patches. This method outperformed the state-of-the-art BM3D algorithm and is particularly good when the image presents similar geometries or textures. On the other hand, the DCT2net denoiser is a supervised deep-learning method, based on the well-known DCT image denoising algorithm, paired with a deep convolutional neural network (CNN). This method, unlike NL-Ridge, requires training the model parameters with a dataset. The DIP algorithm was also evaluated.

4.2.4.1. The DCT2net method

The DCT2net denoiser consists in estimating a DCT basis for each denoising problem by using a CNN and a specific training data-set. In Figure 14, the architecture of the DCT2net network is displayed. It consists in replicating the DCT equations with the help of convolution filters and other mathematical operations. This gives an interpretable method with only 2 layers, making it more interpretable and more efficient in terms of data storage and computing resources. In Figure 15, the difference between the conventional DCT basis and a basis learned by DCT2net is shown. The DCT2net basis is not necessarily orthogonal, as is its counterpart DCT. The main advantage with DCT2net is that the resulting basis can be adapted to a specific class of images, therefore, obtaining much better denoising results when training images are similar to target images.

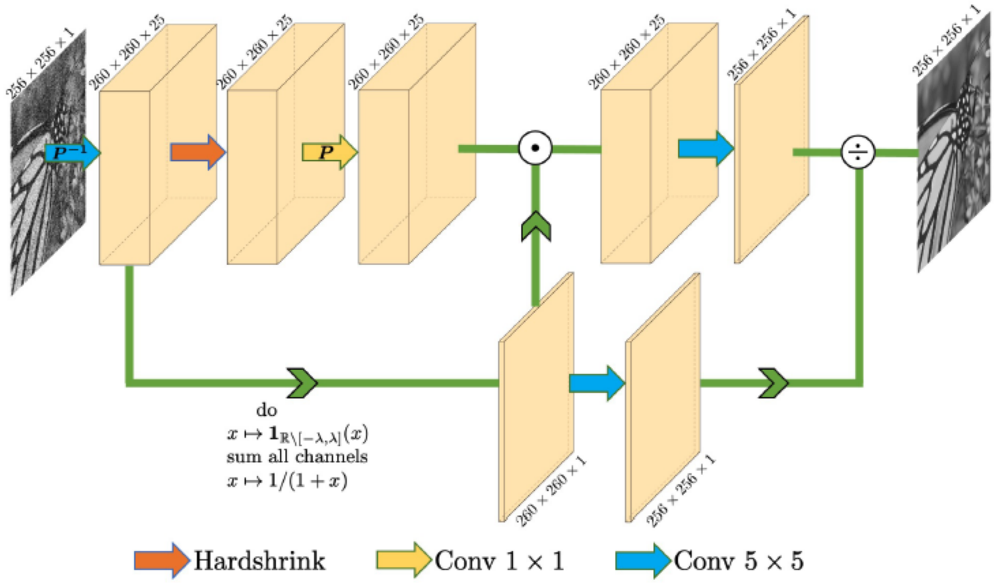


Figure 14: DCT2net: an interpretable shallow CNN (schema)

Because the target images correspond to night satellite images, the SAIRPICO [2] team, with the help of the Institut Curie [4], decided to use microscopy images as a training data-set. This images, provided by the Institut Curie, have very similar characteristics to night satellite images (see Figure 16), which makes them a good candidate for the training procedure. Unlike satellite images, it is possible to control the signal-to-noise ratio in microscopy images by adjusting the light power.

For the following experiments, DCT2net was trained with 20 fluorescence microscopy images of size 1200×1200 pixels, and 2 channels (red and blue) treated separately. This images were trained with simulated additive white Gaussian noise with standard deviation between 0% and 22%.

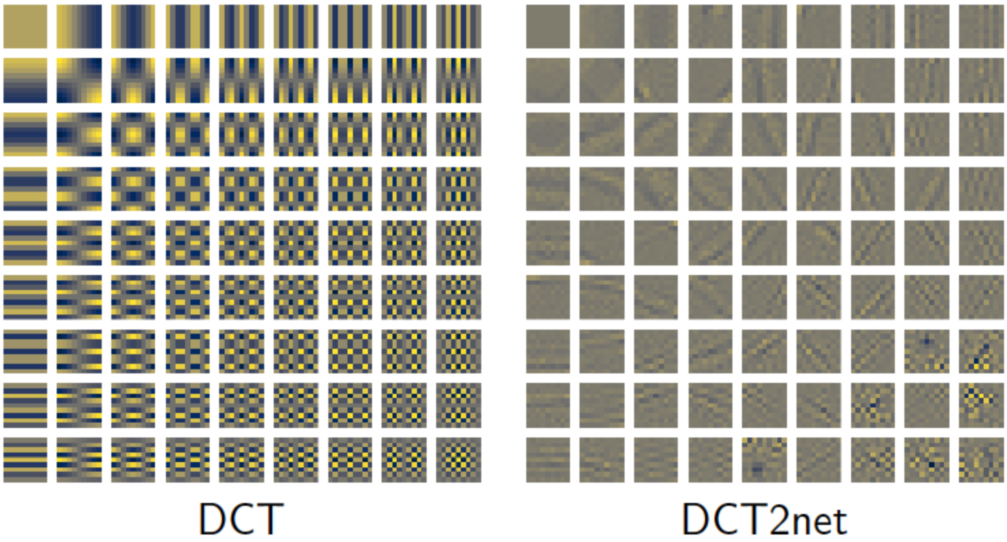


Figure 15: DCT and DCT2net basis comparison

4.3. Experiments

4.3.1. Peak-Signal-to-Noise Ratio (PSNR)

The PSNR value, as explained previously, measures the ratio between the maximum possible power of a signal and the power of corrupting noise. PSNR was defined in (16) via the mean squared error (MSE) and the maximum pixel value of the image. However, in the case of the night satellite images, the datatype associated is *16-bit unsigned integer*. The maximum possible pixel value of this datatype was too large (65535) in comparison to the maximum pixel value of our (dark) target image (between 294 and 331). Therefore, we chose to set MAX_X in (16) to the closest larger power of 2 value to the highest pixel value of the GT image X as follows:

$$MAX_X = 2^{\alpha}, \text{ with } \alpha \in \mathbb{N}^{+} \text{ such that } MAX_X \geq \max(X(i,j)) \; \forall i,j \tag{34}$$

4.3.2. Experiment 1: Denoising

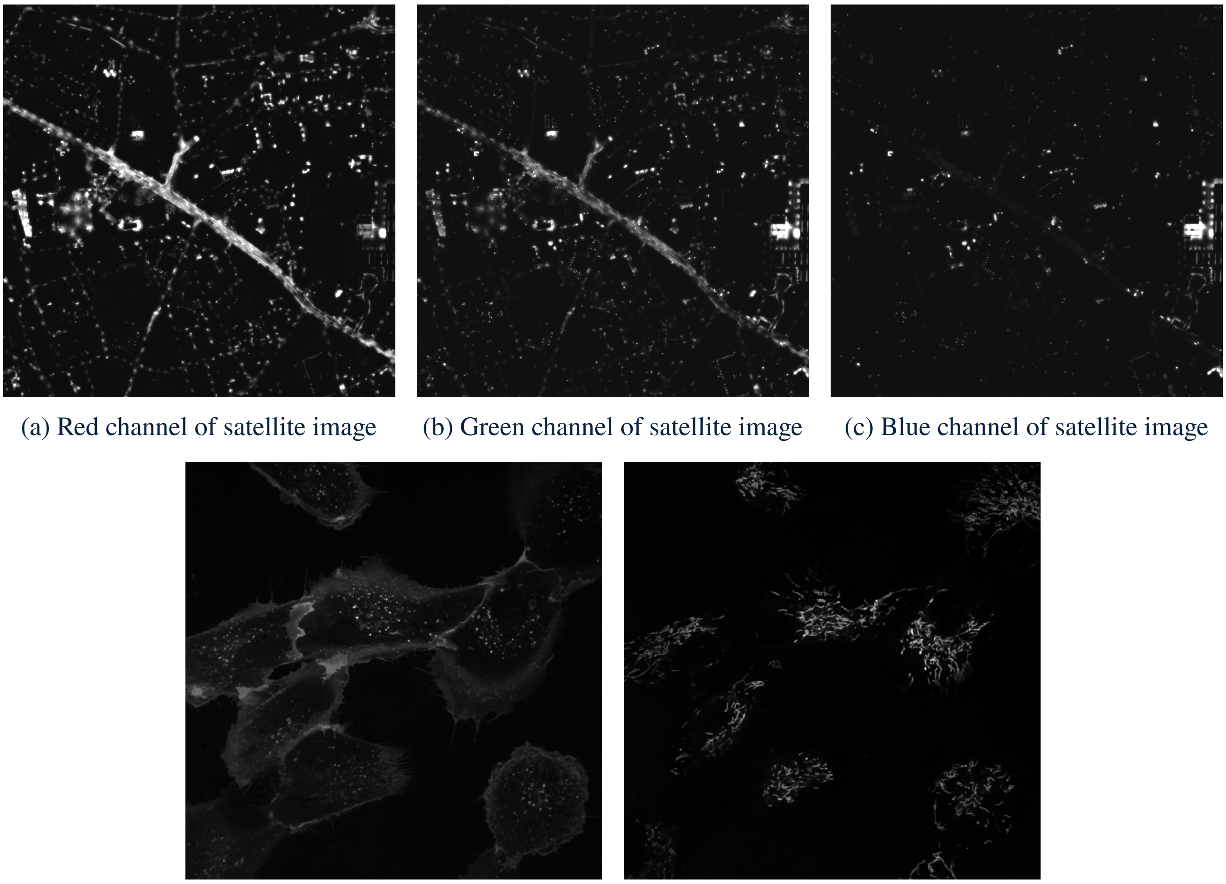
The first experiment consisted in evaluating the denoising capabilities of multiple denoisers applied to night satellite images. First, three 512×512 samples were taken from a large satellite image to make the processing easier and faster (see Figure 17). Then, Gaussian noise was added to each sample image. For $\sigma = 5$, the noisy image for Sample 1, as well as its ground truth, are illustrated in Figure 18.

The denoisers that were analysed were DCT2net [8], NL-Ridge [9] and Deep Image Prior (DIP) [1]. The results in terms of PSNR values are reported in Table 3. It turns out that the PSNR from DCT2net is highest but being very close to NL-Ridge. It is also noted that DCT2net was twice as fast with the architecture used to perform the experiments. However, this difference can be much larger with a better architecture [8], where DCT2net would be 10 times faster. DIP is put just for comparison, as it was thoroughly studied in the first part of this work. However, its performance is considerably lower than the other two methods.

Visually, the results for DCT2net are shown in Figure 19. In Figure 20, the inverted residual image, between the output and the ground truth, are also shown. The more white the pixel, the closer the output is to the ground truth. The areas of the output where the light is highest are the pixels where the denoising performance is worse. On the other hand, the dark background is much better denoised, being closer to the ground truth than the bright areas.

4.3.3. Experiment 2: Deconvolution

The second experiment consisted in evaluating the restoring capabilities of different deconvolution algorithms, notably: Richardson-Lucy [10][11] and Plug-and-Play. As a starting point, different night satellite images



(a) Red channel of satellite image (b) Green channel of satellite image (c) Blue channel of satellite image

(d) Red channel of microscopy image (e) Green channel of microscopy image

Figure 16: Night satellite image (top) and microscopy image (bottom) separated in their corresponding color channels for comparison

were artificially degraded by performing a convolution with a point spread function (PSF) and adding Gaussian or mixed Poisson-Gaussian noise. Then, this images were processed with different deconvolution techniques.

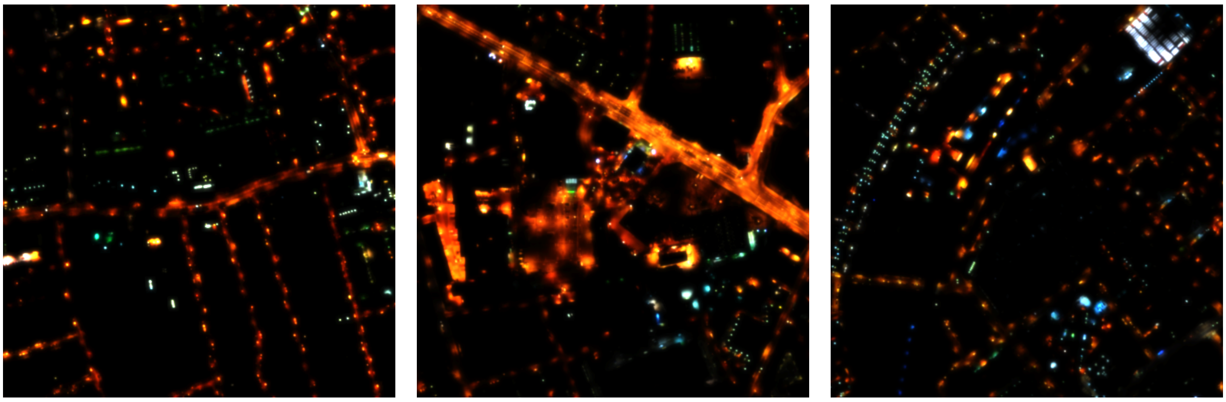
4.3.3.1. The Richardson-Lucy deconvolution method

The Richardson-Lucy algorithm was evaluated on blurry noisy images, and with blurry and noise-free images. The results are displayed in Figure 21. As this algorithm was designed for images corrupted with Poisson noise, it has a good performance both when there is no noise and when it is Poisson distributed (Figure 21c). However, its performance drops considerably when the added noise is Gaussian. We can notice lot of artifacts all around the image (Figure 21d).

4.3.3.2. The Plug-and-Play restoration method

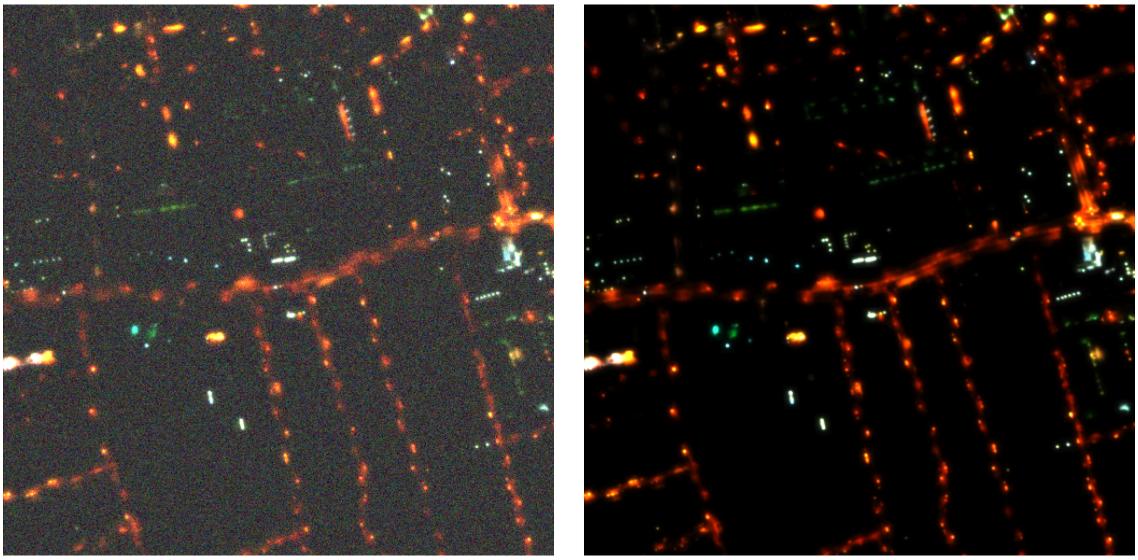
The Plug-and-Play algorithm was analysed with the same set of images as the Richardson-Lucy algorithm. The denoiser used was DCT2net, as it obtained the best results out of all the denoisers tested. Nevertheless, it was necessary to set the parameter λ in order to achieve a good PSNR with as few iterations as possible.

In Figure 22, the PSNR as a function of iterations for different values of λ is plotted. This process was done for various values of λ and the one that performed the best was $\lambda = 10^x$ with $x \in [-2, 3]$ going from the smallest value of the interval to the largest one, linearly, in 50 steps. However, the best PSNR was different for each channel, so it was hard to set a global stopping rule. We considered several rules and we



(a) Sample 1. (b) Sample 2. (c) Sample 3.

Figure 17: Three samples for night satellite images



(a) Noisy (40.08dB). (b) Ground Truth.

Figure 18: Sample 1 with added Gaussian noise of $\sigma = 5$.

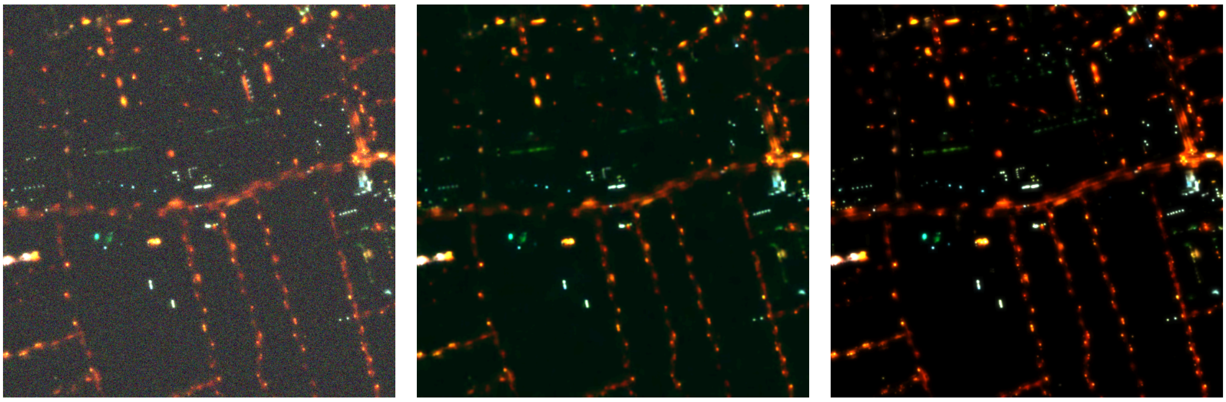
obtained two different modes: a slow one which finishes after 17 iterations (or 13,5 seconds), and a fast one which converges after 8 iterations (or 7,0 seconds). The fast one has a $\lambda = 10^x$ with $x \in [-2, 3]$, like before, but now goes from the smallest value of the interval to the largest one, linearly, in 15 steps instead of 50 (Figure 23).

The difference in PSNR values between these two modes, for Poisson noise, is reported in Table 4. The same comparison but for Poisson-Gaussian noise is given in Table 5. This brought to light that the difference between using the *fast* λ rule and the *slow* λ rule is more significant when processing Poisson-distributed images, in comparison with processing Poisson-Gaussian-distributed images, where the difference is much smaller. Then, the choice of λ depends on the type of noise in the image: when processing images with Poisson noise, the *slow* λ rule was used, and when processing images with Poisson-Gaussian noise, the *fast* λ rule was used.

Once the value of λ was set, the following task consisted in performing the deconvolution to various samples and calculating the PSNR of each sample for comparison.

Table 3: Denoising method comparison for images with Gaussian noise.

Denoising (Gaussian Noise)				
PSNR	Noisy Image	DCT2net	NL-Ridge	DIP
Sample 1	40.08 dB	**51.25 dB**	51.08 dB	47.64 dB
Sample 2	40.14 dB	**51.76 dB**	51.69 dB	46.62 dB
Sample 3	40.15 dB	51.89 dB	**52.25 dB**	47.10 dB
Elapsed Time	Noisy Image	DCT2net	NL-Ridge	DIP
Sample 1	-	0.9 s	2.2 s	1000 s
Sample 2	-	0.9 s	2.3 s	973 s
Sample 3	-	0.9 s	2.2 s	899 s



(a) Noisy (40.08dB). (b) DCT2net Output (51.25 dB). (c) Ground Truth.

Figure 19: DCT2net Denoising: Results for Sample 1

4.3.3.3. Comparison of methods

The results of the Plug-and-Play algorithm were compared with the results produced by the Richardson-Lucy and Deep-Image-Prior algorithms.

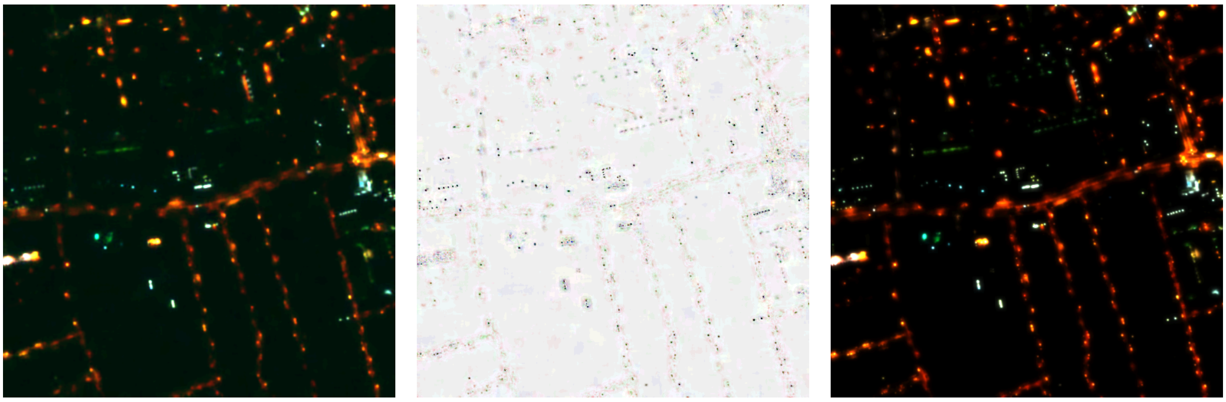
In Table 6, the results for the deconvolution of images with Poisson noise are given. In Table 7, the results for the deconvolution of images with Poisson-Gaussian noise with $\sigma = 5$ are reported.

What is interesting of this results is that the denoiser used with the Plug-and-Play method was trained with only Gaussian noise, not with Poisson noise, while achieving a very good performance regardless. The Deep-Image-Prior algorithm manages to remove the Gaussian noise, as seen in Table 7, but is not able to achieve satisfyingly image deconvolution results. This method was only calculated for one sample, since the time to finish was between 1,5 and 3 hours, and the results were not very promising. The outcome for the other samples would have been very similar to the one calculated. This technique was mainly put here for comparison because it was thoroughly studied in Section 3.

The results of the deconvolution for each method can be visually assessed in Figure 24. The residual images between the noise-free images and the outputs are also shown in Figure 25. These images were

PSNR	Noisy Image	Slow λ	Fast λ
Sample 1	45.58 dB	51.12 dB	50.55 dB
Sample 2	42.36 dB	49.32 dB	48.68 dB
Sample 3	43.31 dB	49.14 dB	48.48 dB

Table 4: Plug-and-Play: PSNR for fast and slow λ modes for Poisson-degraded images.



(a) DCT2net Output (51.25 dB). (b) DCT2net Residual Image. (c) Ground Truth.

Figure 20: DCT2net Denoising: Difference between output and ground truth for Sample 1

PSNR	Noisy Image	Slow λ	Fast λ
Sample 1	39.12 dB	47.02 dB	47.27 dB
Sample 2	38.07 dB	44.90 dB	45.06 dB
Sample 3	38.52 dB	45.25 dB	45.29 dB

Table 5: Plug-and-Play: PSNR values for fast and slow λ rules for images corrupted by Poisson-Gaussian noise.

inverted to clearly detect the zones where the deconvolution performs best. It is clear that the brighter areas in the original image are the ones that perform worst in the deconvolution, whereas the dark zones get almost equal values to the noise-free image.

In the case of the Richardson-Lucy deconvolution (Figure 25d) the performance drops drastically when applying the algorithm to an image corrupted with Poisson-Gaussian noise. However, when it comes to the Poisson noise image, the algorithm performs much better and its residual (see Figure 25a) shows that the background is what it struggles with the most.

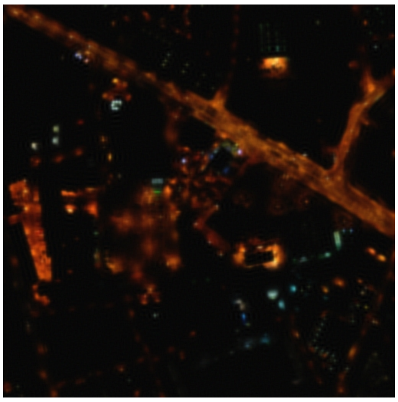
In the case of the Plug-and-Play algorithm with DCT2net as denoiser, the performance is good both in the Poisson and the Poisson-Gaussian case. The major problem for this cases (Figures 25b and 25e) are the brighter areas as previously mentioned.

Finally, in the case of DIP (Figures 25c and 25f), both residuals show that the algorithm is not able to perform the deconvolution but is able to remove Gaussian and Poisson-Gaussian noises.

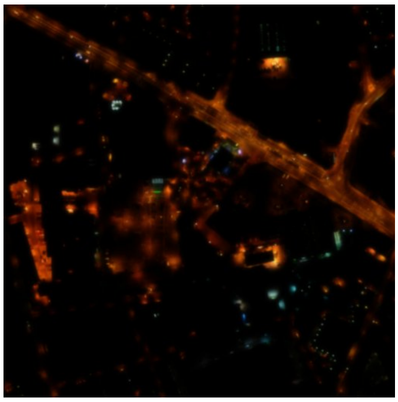
Deconvolution (Poisson Noise)				
PSNR	Noisy Image	Plug-and-Play	Richardson-Lucy	DIP
Sample 1	45.58 dB	**48.21 dB**	47.37 dB	44.72 dB
Sample 2	42.31 dB	**49.29 dB**	45.37 dB	41.29 dB
Sample 3	43.26 dB	**49.10 dB**	46.35 dB	42.91 dB
Elapsed Time	Noisy Image	Plug-and-Play	Richardson-Lucy	DIP
Sample 1	-	13.5 s	0.09 s	~2h
Sample 2	-	13.5 s	0.09 s	~3h
Sample 3	-	13.5 s	0.09 s	~1h

Table 6: Deconvolution method comparison for images with Poisson noise.

(a) Noisy Poisson (42.36 dB)



(c) Deconvolution Poisson in 100 iterations (45.38 dB)



(e) Ground Truth Sample 2

(b) Noisy Poisson-Gaussian (38.16 dB)

(d) Deconvolution Poisson-Gaussian in 5 iterations (38.68 dB)

Figure 21: Richardson-Lucy Deconvolution for different noise types and number of iterations.

4.4. Results Summary

4.4.1. Denoising

The main conclusions from the denoising experience are the following ones:

- DCT2net clearly shows an advantage with respect to the other methods tested. Not only DCT2net tends to produce images with the highest PSNR values, it also is the fastest method being more than twice as fast as NL-Ridge.
- Using microscopy images to train the DCT2net network seems appropriate, since the results show a very good denoising capability.
- The DIP algorithm, as expected, performs way below its counterparts, both in terms of PSNR and, most importantly, in computing time.

4.4.2. Deconvolution

The main conclusions on the deconvolution experience are the following ones:

- Plug-and-Play proves to be a very performing method, mostly when it comes to Gaussian-Poisson images, where the other methods perform poorly.
- DCT2net paired with Plug-and-Play proved that even though the model was trained with images corrupted by Gaussian noise, it still achieved a good performance when performing image deconvolution in the case of Gaussian-Poisson noise.

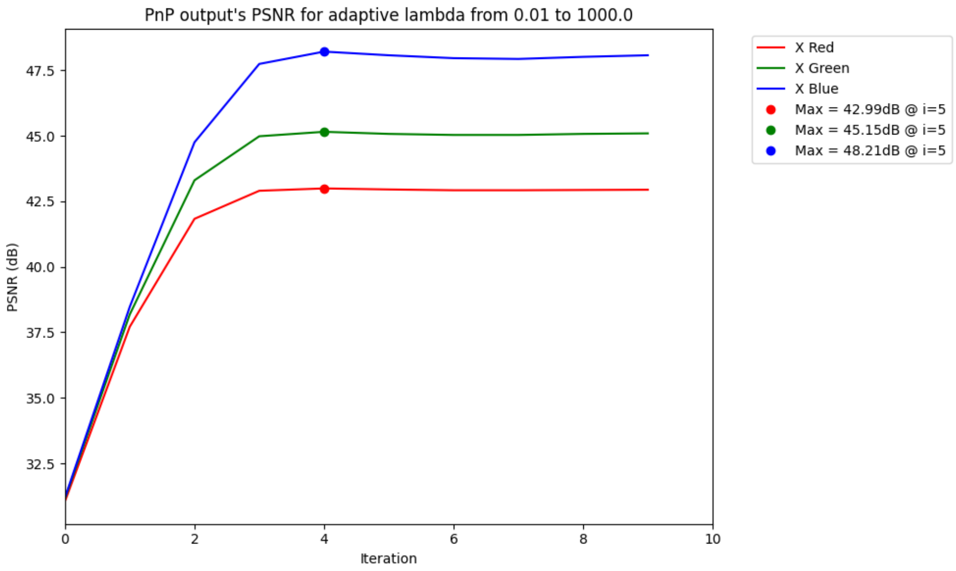


Figure 22: Plug-and-Play: PSNR as a function of iterations for a fast λ.

Deconvolution (Poisson-Gaussian Noise)				
PSNR	Noisy Image	Plug-and-Play	Richardson-Lucy	DIP
Sample 1	39.12 dB	**47.27 dB**	39.15 dB	-
Sample 2	38.07 dB	**45.02 dB**	38.68 dB	41.51 dB
Sample 3	38.52 dB	**45.29 dB**	38.80 dB	42.54 dB
Elapsed Time	Noisy Image	Plug-and-Play	Richardson-Lucy	DIP
Sample 1	-	7.0 s	0.09 s	-
Sample 2	-	7.0 s	0.09 s	~1h30
Sample 3	-	7.0 s	0.09 s	~10m

Table 7: Deconvolution method comparison for images with Poisson-Gaussian noise.
Plug-and-Play with DCT2net as denoiser and fast λ rule (8 iterations).
Richardson-Lucy algorithm without denoiser (5 iterations).

- When it comes to the choice of λ in the Plug-and-Play algorithm, it is better to use an adaptive λ that changes through the iterations, rather than setting λ to a constant value.

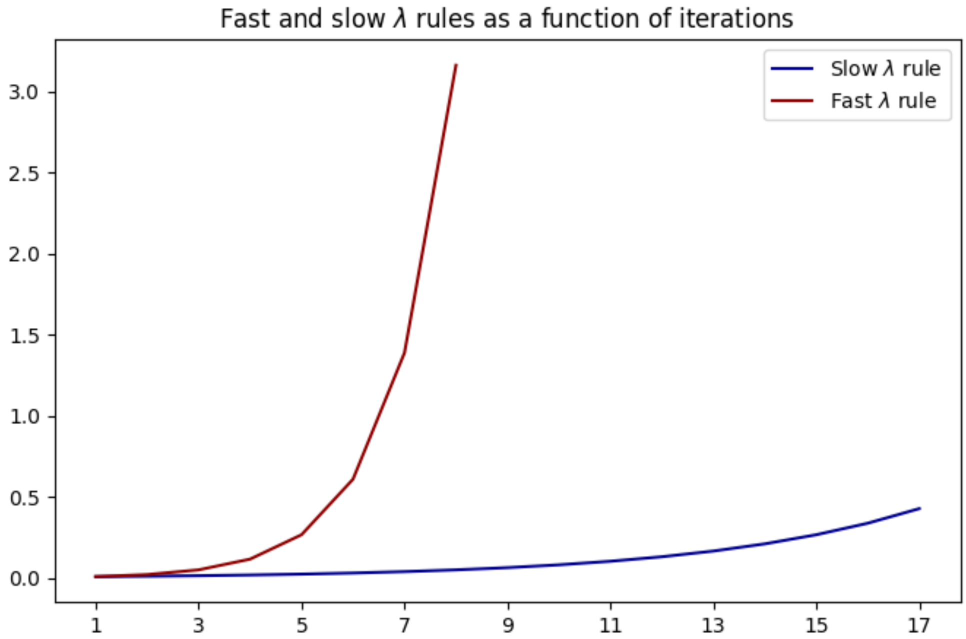


Figure 23: Plug-and-Play: slow and fast λ rules as a function of iterations.

5. Conclusion

In this section, the goal is to summarize the major results obtained in the course of the internship and draw some conclusions. As starting point, the main conclusions from the first experience (Section 3) will be assessed.

- The result of the PCA of the parameters (θ) of the DIP model is as follows: one eigenvector which has almost all the importance (more than 90%), and all the other eigenvectors that are almost irrelevant. However, the result of the PCA of the difference between a vector of parameters and the next one ($\Delta\theta$) gives a bunch of eigenvectors with very close eigenvalues, meaning that from one iteration to the next, the parameters change in such way that no relation can be noticeable.

- A secondary objective, which was to try to generate a new image restoration algorithm from this study, was not achieved. The idea was to be able to use only the gradient descent statistics, and not require the use of a neural network. The problem was that when replacing the neural network with the new designed algorithm, the image restoration procedure stagnated.

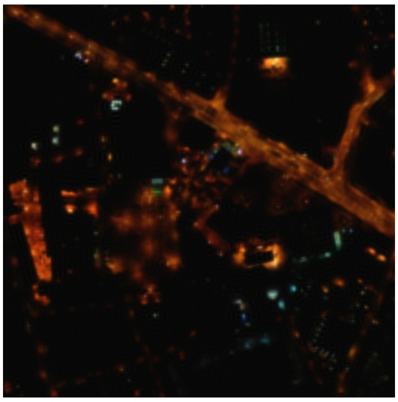
Along with this results, the conclusions for the second experience (Section 4) will be assessed.

- The DCT2net method for denoising is the fastest and, almost always, the best performing method when it comes to denoising night satellite images. It is fast, interpretable, efficient in terms of data storage and computation.

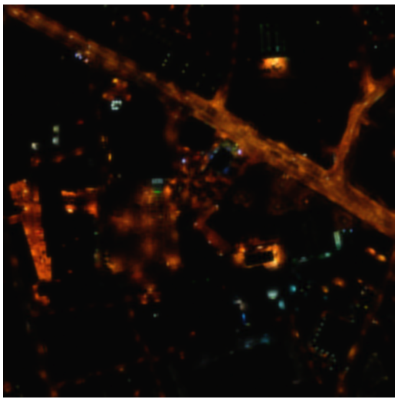
- The Plug-and-Play algorithm, paired with the DCT2net denoiser, proved to perform well with blurry images degraded with both Poisson noise and Poisson-Gaussian noise. It is a method that takes much more time to finish than Richardson-Lucy, but obtains much better results when it comes to images corrupted by Poisson-Gaussian noise.

5.1. Further development of the project

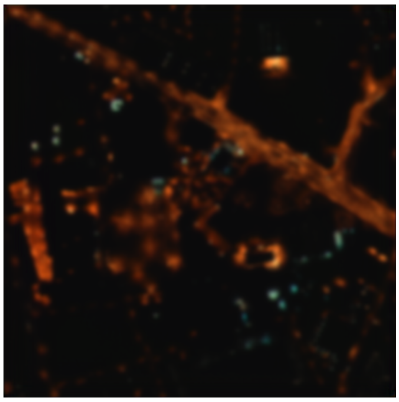
The first part of the internship remains a subject of study and might be continued in the course of the last month of internship. It would be interesting to see if with an approach such as this one, a new disruptive denoising algorithm could be designed. Regardless of its feasibility, it stays very interesting subject of study.



(a) Richardson-Lucy Poisson 100 iterations (45.38 dB)



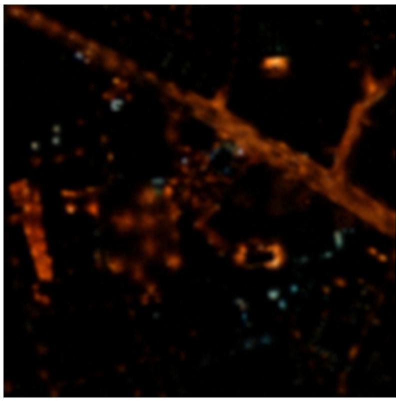
(b) Plug-and-Play Poisson with slow λ rule (49.32 dB)



(c) Deep-Image-Prior Poisson (41.29 dB)

(d) Richardson-Lucy Poisson-Gaussian 5 iterations (38.68 dB)

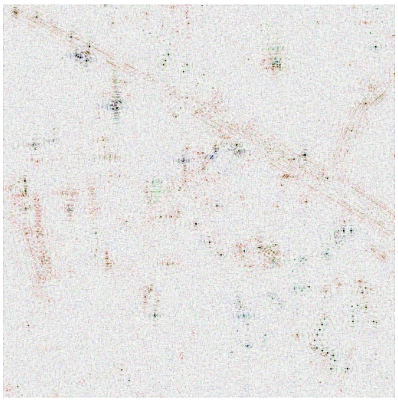
(e) Plug-and-Play Poisson-Gaussian with fast λ rule (45.02 dB)



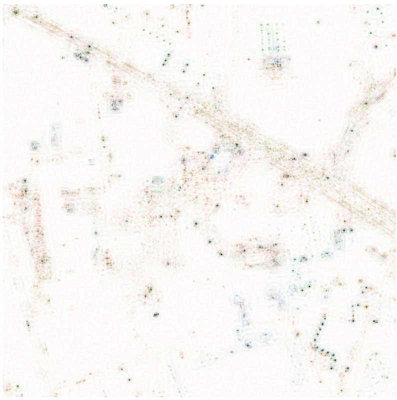
(f) Deep-Image-Prior Poisson-Gaussian (41.51 dB)

Figure 24: Deconvolution method comparison for different noise types: Poisson and Poisson-Gaussian with $\sigma = 5$. Plug-and-Play using DCT2net as denoiser.

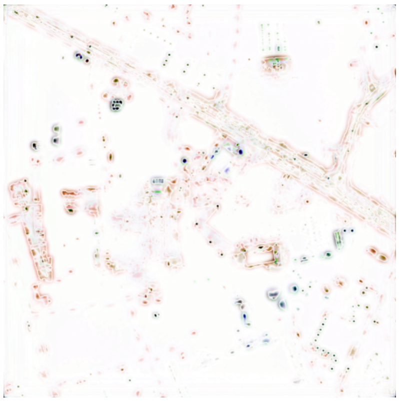
As the internship will continue until the end of September, some other tests will be performed, in particular, the evaluation of more denoising and deconvolution methods. This methods might be tested on different kind of images, such as microscopy images, and their performances evaluated. Re-training the DCT2net algorithm may be performed with a more dedicated dataset.



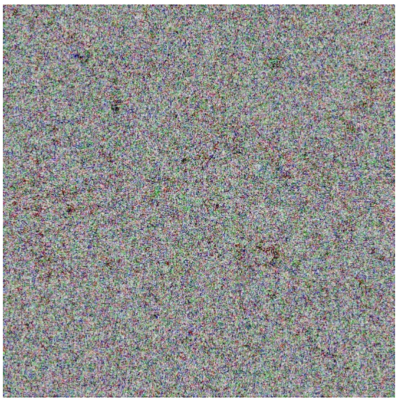
(a) Richardson-Lucy Poisson 100 iters. (45.38 dB)



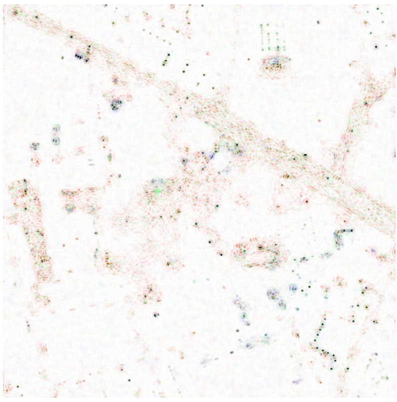
(b) Plug-and-Play Poisson with slow λ (49.32 dB)



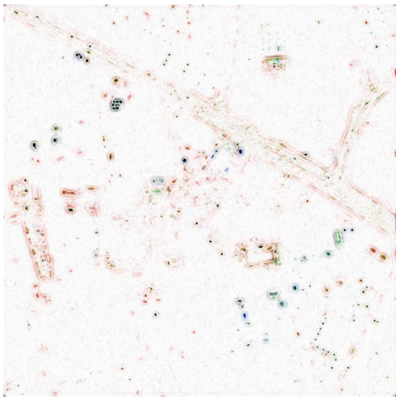
(c) Deep-Image-Prior Poisson (41.29 dB)



(d) Richardson-Lucy Poisson-Gaussian 5 iters. (38.68 dB)



(e) Plug-and-Play Poisson-Gaussian with fast λ (45.02 dB)



(f) Deep-Image-Prior Poisson-Gaussian (41.51 dB)

Figure 25: Residual images from deconvolution methods comparison for different noise types: Poisson and Poisson-Gaussian with $\sigma = 5$. Plug-and-Play using DCT2net as denoiser.

6. Project in perspective

This project synthesizes all the work carried out in these last 5 months in the SAIRPICO team at Inria Center of the University of Rennes. It is also the culmination of six years of studies, started at Universidad de la Republica in Uruguay in 2018, and finished in 2023, at the French "grande école" IMT Atlantique. I am very proud to have completed this process and I feel that I have obtained the necessary tools to move confidently in the future.

In the following lines, I would like to reflect on the internship, on its possible impacts in different fields, on the different skills and knowledge I acquired and their complementarity with my academic path.

6.1. Development of the internship

This internship was carried out in three main phases. First, a bibliographic stage where I informed myself about the subject in depth. Secondly, I focused on the first objective, extracted results and conclusions and documented the procedure in detail. Then, as part of this same second stage, I went on to work on the second objective, where I had to incorporate new bibliographic readings. Once this second phase was completed, I focused on documenting all the work done, the results, the conclusions, and putting it into perspective.

6.1.1. Bibliographic study

In this first approach to the problem, I was able to become immersed in the problem thanks to the reading of several scientific articles and the communication with my tutor, as well as with my other work colleagues: graduate students, postgraduate students and engineers. I was fortunate to be in a work environment where I was able to express my doubts and get the necessary guidance from the team. In this first part I was able to see that many of the things I learned during my studies were directly applicable to this project, which allowed me to move with ease and quickly understand many of the problems involved.

6.1.2. Development phase

At this stage, I started to implement the different methods and algorithms previously studied and to empirically observe the results identified during the bibliographic study. The beginning of this process was very enriching, since I managed to familiarize myself with the project, applying what was previously studied, without major problems. To run all the necessary codes and programs, I had a large computational server at my disposal, with which I launched several of my tests on GPU and I benefited from the higher processing speed.

As a starting point, I focused on the initial objective of studying the statistics behind the gradient descent. For this, I studied a specific machine learning algorithm applied to unsupervised image restoration: Deep-Image-Prior. Once I had written a working code of this algorithm, I started to perform different tests on it, as explained in the first part of the methodology (Section 3). It was a very interesting process, since it consisted of adding functionalities and observing intermediate outputs in the code, to then observe possible patterns, which we then analyzed together with the team.

This study had several interesting results and several conclusions were drawn. I believe there is room for further study in the future, but we decided to continue with the second objective and leave this one for another time.

I then switched to the restoration of night satellite imagery (Section 4). To achieve this, I had to read several articles and talk to other members of the team to find out how they had tackled this problem in the past. They were very supportive and gave me good advice on how to proceed. In this part, I managed to test several different algorithms, implementing them in code and coming up with various results. I enjoyed being able to apply both the knowledge acquired during my studies, and what I had learned so far during the internship. As I obtained results, I synthesised some conclusions, and presented them to my team all throughout the internship.

Both the work done in the first part and the second part could be published in a research journal by the team in collaboration with a PhD student.

6.1.3. Documentation

Documentation was carried out throughout the course of the internship. From saving the results in images and tables within a text document, to taking note of the objectives and all the suggestions and comments made during the weekly meetings with my tutor and the team. At the same time, the continuous presentation of results and conclusions throughout my work, allowed me to synthesize the experiences in a modular way, which greatly facilitated the drafting of the final report. In this last stage, in addition to reviewing and wrapping up all the work done, I was able to reflect on the subject, on my performance, my work space, my objectives, and the impact that my work may have in the future.

6.2. Impact of the work

This work has a direct impact on a project being carried out by Airbus, which aims to launch a satellite into orbit and capture night-time images of the Earth. There are several teams within the Inria Center of Rennes University working on this same large project, but in other areas such as robotics.

This will allow the company to obtain a great improvement in the quality of its images, processing them almost in real time and thus carry out their planned operations.

Also, a positive social impact is foreseeable, since the restoration of nocturnal images can be extrapolated to images from microscopes or other medical devices which output images that are degraded in some way.

This would help physicians to make a better, more accurate diagnosis by observing much clearer images.

Finally, from an environmental perspective, this project could lead to the development of an image restoration algorithm that does not require the use of a neural network. This would save a lot of energy, since today, neural network training is responsible for a large part of the energy consumption of servers and computers.

6.3. Skills acquired

This project allowed me to learn a lot, not only technically about a very interesting topic, but also to develop my soft skills within the workspace.

6.3.1. Technical skills

First of all, this project allowed me to deepen my technical knowledge in the area of signal processing, more particularly image processing. I learned several different methods of image restoration and I learned a lot about how to handle the images themselves: the different formats, their histograms and some forms of compression. I became familiar with programs like *ImageJ* and libraries like *scikit-image*. I deepened my knowledge of AI and machine learning, as well as my Python programming skills. I learned new concepts about statistics, and how to apply them to a specific task, turning complex mathematical formulations into code executable by a computer. I also learned new ways to connect to computational servers, which I did not know previously, and which allowed me to run my programs at a faster speed.

6.3.2. Complementary skills acquired at work and at school

This internship also allowed me to apply much of the knowledge acquired in this last year, both in the courses of IMT Atlantique and in the Master of Research at the University of Rennes 1. Some of the courses that I highlight that had a strong impact on my performance were Introduction to Machine Learning, Computer Vision, Computational Imaging, Estimation and Inverse Problems. Besides the technical area, also in the communication area I was able to use my soft skills acquired at IMT Atlantique to my work at Inria. Thanks to the many project presentations I had to do at school, I was able to present my results and conclusions more fluently and in an organized and clear way, which was highly appreciated by my tutor and teammates.

6.3.3. Organisational skills

In addition to the technical skills, I also acquired a lot of knowledge in the area of organization of results and space management. Having had to do many different experiences, I had to keep track of the results I was obtaining and then present them in an orderly fashion. This presented a challenge for me, as I happened to have many files and images on the computer that at a certain point became difficult to organize properly. It forced me to find a clear system of file nomenclature in order to keep order as I kept adding new files. I also had to organize my codes, and store them appropriately as I made progress. For this I used Inria's *GitHub* platform, a tool with which I was already familiar from the projects I did throughout my career. Finally, for the presentations I used *LibreOffice* tools and for the writing of this report I used LaTeX, another tool that I also used regularly during my studies.

6.3.4. Interpersonal skills

Last but not least, I am happy to say that I also acquired several interpersonal communication tools, both in communicating with my boss and with my office colleagues. Every Friday we had a meeting with all the members of the SAIRPICO team, each one with different backgrounds and working on a different project. We all presented, in a summarized manner, what we had done during that week. It was an good exercise, since I had to explain my progress to people who were not familiar with my subject, and also learn about what the others were working on.

In the course of the internship, I did several activities with my colleagues. For example, we did road-trips to visit other areas of Brittany, which strengthened bonds within the team.

The workplace was culturally diverse, which is something I really appreciated. I got to meet people from other countries and had very interesting conversations with them, which allowed broaden my knowledge of other cultures and ways of life.

Overall, I believe that this was an extremely enriching experience, both academically and personally.

7. My professional project

The completion of this internship also indicates the culmination of an arduous but beautiful process in which I learned a lot about my specialization, and about life.

It corresponds to the end of my studies both at IMT Atlantique and at Universidad de la República. I will be obtaining the degree in Communication Systems and the "Diplôme d'ingénieur généraliste", as well as the M2 Master SISEA in image treatment from the University of Rennes. The acquired skills will allow me to be well prepared to continue with my professional project. This internship enabled me to get an insider's view of the world of research. I would love to contribute to the scientific community in the future, possibly pursuing a PhD. Whether in research or industry, what I am certain is that I want to work on a challenging project that has a positive impact on society. I am excited about the future, to continue learning and improving, and happy to be able to pursue what I love.

8. Glossary

Below the list of abbreviations and vocabulary used throughout this manuscript.

- **Inria**: "Institut national de recherche en sciences et technologies du numérique"
- **SAIRPICO**: Space-time imaging, artificial intelligence and computing for cellular and chemical biology
- **DIP**: Deep-Image-Prior
- **ES**: Early-stopping
- **TV**: Total Variation
- **MCMC**: Markov Chain Monte Carlo
- **MH**: Metropolis-Hastings
- **PCA**: Principal Component Analysis
- **PSNR**: Peak Signal to Noise Ratio
- **GT**: Ground Truth
- **PSF**: Point Spread Function
- **CNN**: Convolutional Neural Network
- **DCT**: Discrete Cosine Transform

References

[1] V. Lempitsky, A. Vedaldi and D. Ulyanov, "Deep Image Prior," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 9446-9454, doi: 10.1109/CVPR.2018.00984.

[2] SAIRPICO Project Team. (n.d.) *Space-timE RePresentation, Imaging and cellular dynamics of molecular COmplexes* https://team.inria.fr/serpico/

[3] Inria Center of Rennes University (n.d.) *Inria Center of Rennes University* https://www.inria.fr/en/inria-centre-rennes-university

[4] CNRS Institut Curie. (n.d.). *Institut Curie Research Center*. https://institut-curie.org/page/institut-curie-research-center

[5] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. *Equation of state calculations.* J. Chem. Phys. 21, 1087-1092

[6] Moebel E., Kervrann C., 2020. *"A Monte Carlo framework for missing wedge restoration and noise removal in cryo-electron tomography"*. Journal of Structural Biology: X 4 (2020) 100013. doi:10.1016/j.yjsbx.2019.100013.

[7] Hengkang, W., Taihui, L., Zhong, Z., Tiancong, C., Hengyue, L., Ju, S., 2021. *Early Stopping for Deep Image Prior*, arXiv:2112.06074v2 [cs.CV], https://arxiv.org/abs/2112.06074v2

[8] S. Herbreteau and C. Kervrann, *"DCT2net: An Interpretable Shallow CNN for Image Denoising"*, in IEEE Transactions on Image Processing, vol. 31, pp. 4292-4305, 2022, doi: 10.1109/TIP.2022.3181488.

[9] S. Herbreteau and C. Kervrann, *"Towards a unified view of unsupervised non-local methods for image denoising: the NL-Ridge approach"*, in arXiv:2203.00570 [eess.IV], https://arxiv.org/abs/2203.00570, 2022.

[10] Richardson, William Hadley (1972). *"Bayesian-Based Iterative Method of Image Restoration"*. Journal of the Optical Society of America. 62 (1): 55–59. Bibcode:1972JOSA...62...55R. doi:10.1364/JOSA.62.000055.

[11] Lucy, L. B. (1974). *"An iterative technique for the rectification of observed distributions"*. Astronomical Journal. 79 (6): 745–754. Bibcode:1974AJ.....79..745L. doi:10.1086/111605.

[12] Macé, A.S. (2018). Chapter 4: Maximum a posteriori estimate for deconvolution with positivity constraint. In *"Super-resolution algorithms for laser scanning microscopy with illumination patterns shaped by conical diffraction"*. Université Sorbonne Paris Cité. Français. ⟨NNT : 2018USPCB212⟩. ⟨tel-02468152⟩.

[13] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, *"Plug-and-play priors for model based reconstruction"*, in 2013 IEEE Global Conference on Signal and Information Processing, 2013, pp. 945–948.

[14] D. Geman and C. Yang, *"Nonlinear image recovery with half quadratic regularization"*, IEEE transactions on Image Processing, vol. 4, no. 7, pp. 932–946, 1995.

[15] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[16] Ho, J., Jain, A. & Abbeel, P. *"Denoising diffusion probabilistic models"*. Advances in Neural Information Processing Systems 33, 6840-6851. (2020).

[17] Song, Y. & Ermon, S. *"Generative modeling by estimating gradients of the data distribution"*. Advances in Neural Information Processing Systems 32. (2019).

[18] Dhariwal, P. & Nichol, A. *"Diffusion models beat GANs on image synthesis"*. Advances in Neural Information Processing Systems 34, 8780-8794. (2021).

OUR WORLDWIDE PARTNERS UNIVERSITIES - DOUBLE DEGREE AGREEMENTS

3 CAMPUS, 1 SITE

IMT Atlantique Bretagne–Pays de la Loire – **http://www.imt-atlantique.fr/**

Campus de Brest
Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
T +33 (0)2 29 00 11 11
F +33 (0)2 29 00 10 00

Campus de Nantes
4, rue Alfred Kastler
CS 20722
44307 Nantes Cedex 3
France
T +33 (0)2 51 85 81 00
F +33 (0)2 99 12 70 08

Campus de Rennes
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
France
T +33 (0)2 99 12 70 00
F +33 (0)2 51 85 81 99

Site de Toulouse
10, avenue Édouard Belin
BP 44004
31028 Toulouse Cedex 04
France
T +33 (0)5 61 33 83 65

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom