Self-supervised methods for low level vision

Mohammed HSSEIN
ENS Paris-Saclay
4 Av. des Sciences, 91190 Gif-sur-Yvette, France
mohammed.hssein@ens-paris-saclay.fr

Soufiane FAFE
ENS Paris-Saclay
4 Av. des Sciences, 91190 Gif-sur-Yvette, France
soufiane.fafe@polytechnique.edu

Abstract

Image restoration tasks such as denoising have taken great attention in the field of Image processing. The idea is simple : based on a corrupted image, try to reconstruct a better one. The term ***better*** *here is a bit vague, and it is key since it measures the performance of the denoising task. In this context, we mean by better an image of high quality to the human eye compared with the initial image. We will restrain our task in this project to the problem of* ***image denoising****. We present a quick recap of the main ideas of papers this project is based on, discuss the experiments we have done, and show the quantitative and qualitative results.*

keywords : ***Image denoising***, ***Deep learning***, ***Neural networks***, ***Self-Supervision***, ***Noise2Noise***, ***Noise2Void***, ***Noise2Self***, ***Blind-spot Networks***

1. Introduction

Image denoising, is the process of cleaning noisy images from noise. Mathematically, given an image (signal) $\mathbf{x}$ obtained with some specific perturbed process, we assume this image have the form $\mathbf{x} = \mathbf{s} + \mathbf{n}$ where $\mathbf{s}$ denotes the true signal/image we would have obtained if the process was perfect, and $\mathbf{n}$ denotes some noise, for examples induced by the production process [2, 1, 5, 3]. In addition we assume that the noise $\mathbf{n}$ is zero-mean, which means that if we generate by the same mechanism, $\mathcal{D} = \{\mathbf{x}_i = \mathbf{s} + \mathbf{n}_i\}_{1 \leq i \leq n}$ images, and we average them arithmetically, we would approach the true value of the pure signal/image $\mathbf{s}$. The goal of image denoising, is then to extract as much as possible the clear signal $\mathbf{s}$ from the image $\mathbf{x}$. Traditionally, methods like **non local means**, **median filter**, **block matching & 3D filtering** have been state of the art methods in performing such tasks for a while. However, with the tremendous development of neural networks, image denoising has become dominated by deep learning approaches. Contrary to classical methods, deep learning relies on *learning* to denoise, by looking in a **supervised** way at thousands or millions of examples, and thus capture more information to handle serenely the denoising task. However, the main difficulty with such approaches, is the lack of pairs of **clean/corrupted** data making discriminative deep learning methods hard to exploit. In addition, a classic issue with classic methods, is the need to model the prior $p(\text{noise}|\text{clean})$ or $p(\text{clean})$. Known distribution as the multivariate Gaussian, are usually used. However, when the noise of the image is from an unkown type, the performance of such methods is expected to degrade. Moreover, it is experimentally shown [4] in the field of image analysis, that the signal contained in image patches, contains more relevant information compared to information coming from external patches of other images, and thus **internal statistics**, could bring more predictive power to image denoising tasks with deep learning. These ideas motivate considering new methods that can get rid of the need to huge quantity of data, and combine the predictive power of deep neural networks, with the internal statistics contained insides patches within a single images. In the following we will have a quick review on three such methods. In the sequel, we will discuss the principles behind these methods in the next section 2, and then we will discuss the practical implementations in section 3, and finally we end this report by analyzing the experimental results we have got in the last section 4.

2. Noise2Something methods

2.1. Noise2Noise approach

The first method we have seen is called **noise2noise** [1]. It belongs to the family of discriminative deep learning. However, this technique is not a modification of the architectures, but only brings a new paradigm of training deep nets by telling us we can omit the need of clean/noisy pairs for supervised training, and use only corrupted images for learning. The ultimate goal of learning using deep nets as function approximation f_θ in statistical learning theory, is to solve:

$$\bar{\theta^*} = \text{argmin}_{\theta \in \Theta} \, \mathbb{E}_{(\mathbf{x},\mathbf{y})} \mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y}) \tag{1}$$

which is equivalent to:

$$\bar{\theta^*} = \text{argmin}_{\theta \in \Theta} \, \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{y}|\mathbf{x}} \mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y}) \tag{2}$$

and reduces in practice to minimize only the empirical version of this risk. We observe that one can solve for each x the problem $\text{argmin}_{\theta \in \Theta} \mathbb{E}_{\mathbf{y}|\mathbf{x}} \mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y})$ which depends only on the law $\mathbf{y}|\mathbf{x}$. Thus if we change the pair $(\mathbf{x}, \mathbf{y}|\mathbf{x})$ with another pair $(\mathbf{x}', \mathbf{y}'|\mathbf{x}')$ the learned parameters of the network will remain the same. Therefore, if we train our network using independently corrupted pairs of images, the parameters learned shall remain the same. This approach has many advantages:

- We do not need any clean image for training, and there is no constraint on the topology of the neural network used. It can be any well-known architecture as for example a U-Net.
- Looking at equation 2, we can see we need no additional information on the noise $p(\text{noise}|\text{clean})$ nor the signal model $p(\text{clean})$ to solve the optimization problem.

2.2. Noise2Void approach

Despite all the advantages of the 2.1 method, this approach still have these shortcomings:

- Noise2Noise training, requires the availability of pairs of noisy images, while the acquisition of such pairs with quasi constant signal of interest $\mathbf{s}$ is only possible for quasi static scenes, which is not achievable in general in many real applications as for instance in biology.
- Noise2Noise training, does not rely on internal statistics of images, but rather on external information.

That is what authors in **noise2void** [3] have tried to solve successfully. They solved the previous shortcomings while leveraging on the same idea of training without need of clean data. Noise2Void is an **internal statistics** method. Its first major good point is that It can enable the training at inference time, i.e it makes it possible to train the network on a single test image using the information contained inside patches of the same image. The authors, have pushed forward the previous idea of **blind denoising** (i.e randomizing the noise level for each training pair to avoid learning one noise level), by introducing the concept of **blind spot networks**. In Noise2Noise 2.1 training, the pairs of training data have the form $\mathbf{x}_j = \mathbf{s}_j + \mathbf{n}_j$ and $\hat{\mathbf{x}}_j = \mathbf{s}_j + \hat{\mathbf{n}}_j$. Here the idea is to use a patch from either single noisy image (or a set of images in the descriminative mode training) as the input, and only its central pixel value as output. However, with this setting, the network might learn the identity function by mapping the square to its central value simply. To overcome this issue, the idea of **blind spot** is introduced and means we will somehow **hide** the central pixel of the input and try to predict its value as output, using only information available in its surrounding pixels. Practically, implementing this idea naively, would not be efficient at all. Indeed, we have to compute in the back-prop step many gradients to predict only the value of one pixel only! The idea in practice, is to extract patches of a certain size (here 64×64) bigger than the receptive fields, then select N pixels within each patch using **stratified sampling**. We mask them to constitute the input images, and try to teach the network to reproduce these N values instead of one value each iteration. But why the network in this case cannot only learn the identity ? In addition of assuming the signal is not pixel-wise independent, which is a fair hypothesis, another hypothesis allow to justify the last fact. The noise is supposed to be **pixel wise independent given the signal**. With this hypothesis, the pixels surrounding the central pixel, carry no information about $\mathbf{n}_i$, which makes it impossible to the network to produce something better than the apriori expected value of $\mathbf{n}_i$, that is $\mathbb{E}[\mathbf{n}_i] = 0$. In practice, instead of masking the value of the central pixel, we rather replace it with a random value from one of the pixels in its neighbor inside the training input patch.

2.3. Improved version of Noise2Void

In High-quality self-supervised deep image denoising, [2], the authors, have proposed to push forward this idea of Noise2Void training 2.2, to give it more information at test time, but with the cost of assuming a specific model for the prior of clean images given their context (surrounding pixels) $p(\mathbf{x}|\Omega_\mathbf{y}) \sim \mathcal{N}(\mu_\mathbf{x}, \Sigma_\mathbf{x})$. The algorithm then works in two steps at inference time, It first predicts the value of $(\mu_\mathbf{x}, \Sigma_\mathbf{x})$, and then computes $\mathbb{E}_\mathbf{x}[\mathbf{x}|\mathbf{y}, \Omega_\mathbf{x}]$.

3. Experiments

The method we have chosen to focus on is Noise2Void 2.2, [3]. As the authors propose an implementation using

TensorFlow 2, we started from their code available at the GitHub repository [1]. To compare with supervised learning methods, we used two among the state-of-the-art models designed for image restoration, based on GANs and transformers respectively. The code for using those models is available at the GitHub repository: [2]. To compare with classic methods as **BM3D**, already implemented in the package **bm3d** in python, and ready to install using python package installer pip via `pip install bm3d`, we wrote a simple jupyter-notebook to corrupt images of BSD68 test data, Kodak, and Set14 with Set5 respectively using Gaussian noise with mean 0, and $\sigma = 25$. We then wrote simple functions to compute some useful metrics as peak signal to noise ratio **PSNR** and structural similarity **SSIM**. We did many experiments using the previously mentioned dataset **kodak**, **Set14**, **BSD300** (with 68 images for test called BSD68 in our report!), and **Set5**. The first experiment we have done, consists on reproducing the main results of the main paper [3] by training with an old fasion way on an entire database of images BSD300. The network architecture used is U-Net. We trained this architecture on 300 images (with N2V paradigm described in 2.2), and tested it on 68 never-seen images. We corrupted the test images by using the same Gaussian noise value, ie a standard deviation of $\sigma = 25$. We compared here our U-net trained with noise2void technique, with the classic BM3D algorithm, and with the 2 other supervised learning models already trained on larger datasets. The results of these comparisons are in **??**. Finally, we tried to train the U-net network only on single images at test time, and compare with BM3D algorithm. The images we used were downloaded from the website Flicker [3]. The results are in 4. The training on large dataset BSD300 took 5 hours on a GPU. However, the training and inference using only one single images takes only 30 to 40 minutes per minutes depending on the size of the image and has lead to very impressive results (we used very large images), and this is a good point about these approaches.

4. Results

The results are shown in the following tables. Our model is slightly less performant compared with the other models. This is in some sort logical, because at test time we do not use any additional information related to the distribution of clean image. This point is solved by [2]. However, for training on one single image at test time, our model is much more performant compared to BMD3 for example.

Models	BM3D	N2V
Avg PSNR BSD	28.42(±2.39)	27.73(±2.96)
Avg SSIM BSD	0.56(±0.13)	0.54(±0.12)
Avg PSNR Kodak	29.45(±1.97)	27.89(±2.87)
Avg SSIM Kodak	0.80(±0.04)	0.79(±0.05)
Avg PSNR Set14	28.77(±1.99)	27.23(±2.80)
Avg SSIM Set14	0.81(±0.06)	0.79(±0.07)
Avg PSNR Set5	29.87(±1.05)	28.68(±1.32)
Avg SSIM Set5	0.83(±0.07)	0.82(±0.08)

Models	BSRGAN	SwinIR
Avg PSNR BSD	28.50(±0.06)	28.51(±0.07)
Avg SSIM BSD	0.45(±0.11)	0.47(±0.12)
Avg PSNR Kodak	32.47(±1.41)	31.85(±1.25)
Avg SSIM Kodak	0.80(±0.04)	0.80(±0.04)
Avg PSNR Set14	31.97(±1.82)	31.27(±1.61)
Avg SSIM Set14	0.80(±0.04)	0.79(±0.07)
Avg PSNR Set5	32.45(±0.31)	31.45(±0.62)
Avg SSIM Set5	0.81(±0.07)	0.82(±0.08)

Table 1. Our model trained on BSD300 and tested on 4 datasets

Models	BM3D	N2V
PSNR longBeach1	28.56	36.02
SSIM longBeach1	0.34	0.9
PSNR longBeach2	27.87	34.51
SSIM longBeach2	0.5	0.82

Table 2. Results of training on single large images

References

[1] Jaakko Lehtinen Jacob Munkberg Jon Hasselgren Samuli Laine Tero Karras Miika Aittala Timo Aila. Noise2noise: Learning image restoration without clean data, 2018. https://arxiv.org/pdf/1803.04189.pdf. 1, 2

[2] Samuli Laine Tero Karras Jaakko Lehtinen Timo Aila. High-quality self-supervised deep image denoising. https://research.nvidia.com/sites/default/files/pubs/2019-12_High-Quality-Self-Supervised-Deep//laine2019denoising_paper.pdf. 1, 2, 3

[3] Tim-Oliver Buchholz Alexander Krull Florian Jug. Noise2void - learning denoising from single noisy images, 2019. https://arxiv.org/pdf/1811.10980.pdf. 1, 2, 3

[4] Michal Irani Maria Zontak. Internal statistics of a single natural image, 2018. https://www.wisdom.weizmann.ac.il/~vision/SingleImageStatistics/Zontak_Irani_CVPR2011.pdf. 1

[5] Joshua Batson Loic Royer. Noise2self: Blind denoising by self-supervision, 2019. https://arxiv.org/pdf/1901.11365.pdf. 1

[1] https://github.com/juglab/n2v

[2] https://github.com/JingyunLiang/SwinIR

[3] https://www.flickr.com/photos/tags/flicker/

Appendix

In the appendix, we show qualitative results of our experiments.

Figure 1. Results of the experiments. From left to right : **Ground truth**, **Test image**, **Ours**, **BM3D**

Figure 2. Results of the comparison with descriminative deep learning models. The first row: **Ground Truth**, **Test img**, **BM3D**. The second row: **Our**, **BSRGAN**, and **SwinIR**

Figure 3. Results of the comparison with descriminative deep learning models. The first row: **Ground Truth**, **Test img**, **BM3D**. The second row: **Our**, **BSRGAN**, and **SwinIR**

Figure 4. Results of training on single image. From top to down: **noisy image** and **Our prediction**