RecVis project report: self-supervised methods for low-level vision

Nathan Levy
ENS Paris-Saclay
nathan.levy@ens-paris-saclay.fr

Arthur Calvi
ENS Paris-Saclay
arthur.calvi@ens-paris-saclay.fr

1. Self-supervised denoising

CNNs are nowadays the dominant approach for image denoising. Traditionally, the models are trained in a supervised way - starting from a noisy input, a loss between the output and the ground truth clean image is penalized. This training technique is referred as Noise to Clean (N2C) [9]. However, in many cases the acquisition of noisy-clean image pairs may be difficult or even impossible. This has motivated the investigation of self-supervised training methods, learning exclusively from noisy data. In this project, we used the Self-Supervised Deep Denoising method [6], which we will briefly introduce.

Self-Supervised Deep Denoising (SSDN) [6] : Let x and y denote respectively the clean and noisy pixel value, Ω_y denotes the noisy context of y which is its receptive field without the central pixel.

The SSDN training method uses the same noisy image as input and target. To avoid learning the identity during training, the central pixel of the receptive field is masked; hence forming a blindspot network. The network learns to generate a Multivariate Gaussian $p(x|\Omega_y) \sim \mathcal{N}(\mu_x, C_x)$ for each pixel by minimizing the negative log likelihood of the training data $p(y|\Omega_y)$ which depends on both the prior $p(x|\Omega_y)$ and the noise model $p(y|x)$.

$$\underbrace{p(y|\Omega_y)}_{\text{Training data}} = \int \underbrace{p(y|x)}_{\text{Noise model}} \underbrace{p(x|\Omega_y)}_{\text{Prior}} dx \tag{1}$$

Namely, the network is learning to map a context Ω_y to a clean pixel parameterized as a multivariate Gaussian (μ_x, C_x). In addition, to use the noisy measurement y of pixel x, a Bayesian inference is performed. In practice, it is done by using the posterior mean $\mathbb{E}_x[p(x|y, \Omega_y)]$ minimizes the MSE loss and therefore maximizes the PSNR [6].

2. Contributions:

We aim to reproduce the ablation study of the Bayesian inference of the SSDN paper and extend it to real noisy images. First, Arthur reproduced the ablation study on Kodak, Set14 and BSD300 datasets from the SSDN paper for an Additive White Gaussian noise (AWGN) using the official implementation [5]. Nathan ran this same implementation to compare the SSDN training schemes with Noise to Clean (N2C). After this, we extended the comparison to real noisy data. Arthur modified the code in order to evaluate already noisy images and correctly compute the PSNR. In addition, he evaluated several noise models on images from smartphone cameras [1]. Finally, Nathan conducted a series of experiments to illustrate the inherent limitations of the SSDN method.

3. Comparisons

3.1. Metrics

All the papers cited use the PSNR metric to rate the denoising. We decided to complete the study by using the SSIM metric [8] which aim at giving a better indication of the visual result. This index is based on three key components : the contrast, the luminance and the structure of the image. The index ranges between 0 (very different) and 1 (perfect similarity). As it is bounded, it allows an easier interpretation.

3.2. Baselines

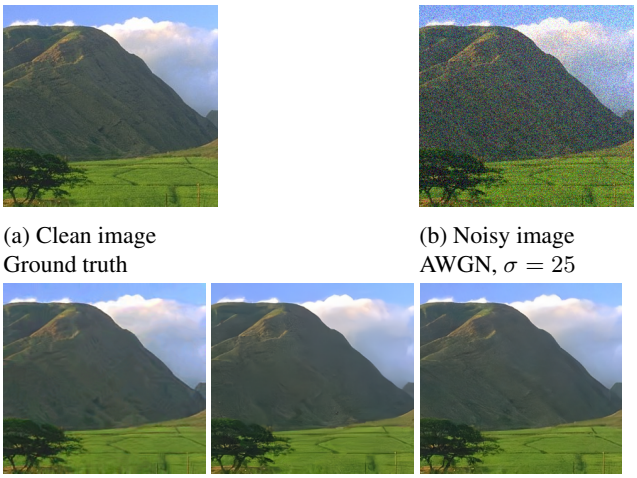
In this section we compare the SSDN performance for denoising AWGN of variance $\sigma = 25$ with the supervised training method Noise To Clean [9], which is also part of the official implementation [5]. We also considered as a baseline the BM3D algorithm, a state-of-the-art non-trained algorithm, implemented in [7].

The results are summarized on Figure 1. For all methods, it is difficult to preserve the mountain grainy structure. The results are fairly close both visually and quantitatively.

4. Ablation study

4.1. Artificial noise

In this section, results of the SSDN paper [6] are reproduced. Precisely, the evaluation on Set14, BSDS300 and Kodak datasets with synthetic white Gaussian noise with



(a) Clean image
Ground truth

(b) Noisy image
AWGN, $\sigma = 25$

(c) BM3D
PSNR=32.71dB

(d) N2C
PSNR=32.91dB

(e) SSDN
PSNR=32.78dB

Figure 1: Performance comparison for Gaussian denoising. Clean image extracted from BSD300.

$\sigma = 25$ is performed. Results are shown in table 1. Three configurations have been tested : with σ known, σ evaluated as a parameter during training and μ only which corresponds to ssdn without the Bayesian inference.

method	PSNR	SSIM
ssdn (σ known)	31.52 dB	0.876
ssdn (σ eval)	31.50 dB	0.875
ssdn (μ only)	29.54 dB	0.820

Table 1: Averaged PNSR and SSIM on Set14, Kodak and BSD300 for a Gaussian noise ($\sigma = 25$)

Results show that the network with the Bayesian inference performs better : on average, PSNR ($+2$dB) and SSIM ($+0.05$) are higher. When σ is known, the results are slightly better. Figure 2 shows the output of the networks. Visually, the images denoised without the Bayesian inference lack of details and sharpness (especially for the coral). Diamond artifacts are visible for the version without the Bayesian inference. This phenomenon is also described in both SSDN and N2V papers; section 5 bring some details on their formation.

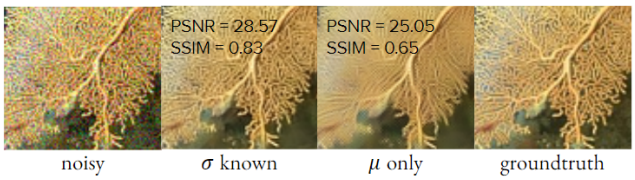


Figure 2: Comparison on artificial noise, Gauss $\sigma = 25$.

4.2. Real noise

The noise model for Raw images is described as :

$$I_{noise} = f([g(Poisson(u(x) + d(x))] + N_R) + N_Q) \quad (1)$$

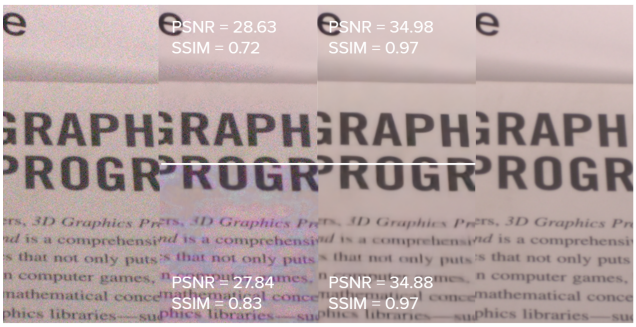
Where f is camera transfer function, g camera global gain, $u(x)$ intensity value at pixel x, $d(x)$ dark current bias, $N_R = \mathcal{N}(\mu_R, \sigma_R)$ reading noise, N_Q uniform noise of quantification. For consumer cameras, one may assume that f is linear, N_Q negligible toward reading noise and dark current negligible for short exposition times, thus the above equation simplify to a Poisson-Gaussian noise :

$$I_{noise} = gPoisson(u(x)) + \mathcal{N}(\mu_R, \sigma_R^2) \quad (2)$$

The Poissonian component models (signal-dependent) the photon-counting process, while the Gaussian accounts for electric and thermal noise [3]. The noise becomes even more complex when working on those images after in-camera processing (e.g., demosaicking, Gamma correction, and compression). This processing increases the complexity of noise and makes it spatially and chromatically correlated [4].

The first experiment consists in denoising central crop 500×500 px of very large RGB images with Poisson and Gaussian noise models to see how the algorithm reacts to a more complex noise. Data are collected from smartphone cameras using CMOS sensor. For the same scene, images are taken from several smartphones and several parameters (ISO and time exposure are varying). Firstly, networks that have to estimate the parameter σ per image fails to denoise. However, if the value of this parameter is fixed in advance (or estimated as a global constant over the entire dataset), the algorithm is able to partially denoise the image even if the noise model does not match the real noise. Figure 3, shows the output of the networks for a Gaussian noise model and a Poisson noise model on a same image taken with a CMOS sensor, ISO800.

Networks without the Bayesian inference are able to give a very decent result. However, as the noise model is incomplete, the Bayesian inference bring erroneous information. This leads to a partial denoising : only areas with low signal are denoised on figure 3 (column: full). The algorithms running with the Poisson noise model achieves the denoising of a larger area of the image. This means that the noise model is not evenly distributed over the image, as it is hinted by equation 2. Focusing on the full algorithm, one can notice that the Poisson model has a higher SSIM but a lower PSNR than the Gaussian model. As the ISO are relatively high, the shot noise (Poisson) may prevail over the reading noise (Gauss). Hence, the Poisson noise model is able to denoise larger areas of the image and then have higher structural similarity with the groundtruth, explaining the higher SSIM. However, other noise convoluted to the



noisy full μ only groundtruth

Figure 3: Comparison on real noisy data. Top : Gaussian model ($\sigma = 25$), bottom : Poisson model ($\lambda = 30$). Camera parameters : ISO800, 1/1000s.

Poisson noise may cause these chromatic artifacts, which lower the PSNR. Indeed, the noise is surely spatially and chromatically correlated due to In-Camera-Processing such as demoisaicking or gamma correction (f not linear). Previous artifacts could be avoided by directly denoising the RAW images as it suggested by Brooks *et al*. for supervised deep learning tasks [2] and implementing the noise model described in equation 2 as a heteroscedastic Gaussian.

	P30		G25	
method	SSIM	PSNR	SSIM	SSIM
ssdn, (σ known)	0.652	22.45 dB	0.639	22.8 dB
ssdn, (σ eval)	0.62	21.12 dB	0.65	23.0 dB
ssdn (μ only)	0.682	23.2 dB	0.707	23.82 dB

Table 2: Comparison of noise models for denoising SSID. P30 : trained with Poisson noise $\lambda = 30$, G25 : trained with Gaussian noise $\sigma = 25$

Table 2 illustrates the results of the 3 configurations of SSDN with two different noise models. Networks without the Bayesian inference are performing better. As contrary as the images commented above, the Gaussian model is better on average on the dataset. Models where the noise parameter (same for all images) has to be estimated are better than for a fixed parameter in the case of Gaussian noise, the contrary for Poisson noise. These results demonstrate the Gaussian noise assumption is better for denoising this dataset. However, results might vary a lot from one image to another. Indeed, the noise is different for each image. Figure 3 shows that the Poissonian noise is better for this image, for instance.

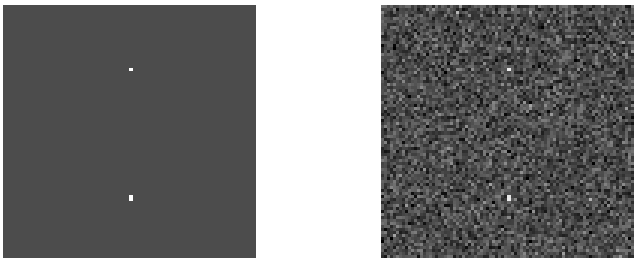
Note : The smartphones tested use CMOS sensor manufactured by Sony ranging from size $1/2.2$" *to* $1/2.7$". *No significant differences are visible after denoising, the denoised areas remain the same for every smartphone. However, the smaller the sensor the higher the level of noise is at the same ISO.*

5. Limitations

Using standard Python packages, we devised a series of experiments showing the limitations of self-supervised training.

5.1. Clean image predictability

One underlying assumption of SSDN is that the clean value x depends on its context Ω_y as well as on the noisy measurement y. The ablated, μ-only SSDN method ignores y and relies only on the context. It outputs the mean μ of the Gaussian approximation to $p(x|\Omega_y)$. For this latter method, the more difficult it is to predict a clean pixel from its surroundings, the more errors are expected. We highlight this limitation by generating a synthetic clean image, consisting of a dark background of size 84x84. One bright isolated pixel is placed on top, and two bright pixels are placed on the bottom. The image is corrupted with AWGN of variance $\sigma = 25$, as shown on Figure 4.



(a) Synthetic clean image (b) Noisy

Figure 4: Synthetic data violating the clean image predictability assumption

Figure 5 summarizes our denoising results for 3 different methods: N2C, SSDN, and the ablated SSDN. The N2C-trained network considers the isolated pixel as pure noise and fills it, whereas the two isolated pixels are interpreted as a structure and kept. The same goes for SSDN. These networks were trained on natural images for which the apparition of a single isolated pixel is very unlikely: in a way, this experiment shows how networks behave when they are confronted with images that do not belong to their learning domain. The ablated version of SSDN yields significantly different results. As expected, it is not possible to denoise an isolated pixel properly as we ignore it. The checkerboard artifact appearing around the isolated pixel illustrates that for blind-spot networks, the operation of denoising a pixel from its surroundings is a kind of inpainting: the initially dark pixels around the bright one become clearer after denoising, as they include information from it.

(a) N2C (b) SSDN (c) SSDN, μ-only

Figure 5: Denoising results on the synthetic image

5.2. Pixel-wise independence of the noise

Another limitation of this method is that it assumes the distribution of noise to be known and that the log-likelihood loss can be computed analytically. Especially, the noise should be pixel-wise independent. To illustrate this drawback, we corrupted a clean image (the grayscale version of Lena) with a structured noise, which has been synthesized in two different ways. To begin with, we added a checkerboard-like strip on top of Lena's hair. This pattern is barely visible when AWGN of variance $\sigma = 25$ is added. The denoising only removes the independent component of the noise, as shown on Figure 6. What happens if the full noise is fully correlated? We built a noise map consisting of the convolution of an AWGN image ($\sigma = 25$) with a Gaussian kernel (various sizes tested, final 7x7). This distribution being different from the theoretical Gaussian model the network was trained with, Figure 7 shows that SSDN completely fails to denoise the corrupted image.

(a) Image corrupted by both AWGN and structured noise (barely visible checkerboard)

(b) Denoised result using the SSDN method. It fails to remove the checkerboard.

Figure 6: Denoising results (checkerboard)

6. Conclusion

We wanted to denoise the raw images of smartphones, but we did not manage to modify the code on time to take into account the right file format. This experiment should have confirmed our hypothesis on the causes of the chromatic artifacts. The ablation study on real noise demonstrates that a more complex noise model is necessary, perhaps a Heteretoscedastic Gaussian. In addition, an auxiliary network has to be trained to correctly estimate the noise parameter per image. The one provided by the authors trained on Gaussian and Poissonian noises failed to estimate the parameters on real data.

(a) Noisy image (b) Denoised image

Figure 7: Denoising results (AWGN noise convolved with a Gaussian kernel

Finally, our experiments highlighted the limitations arising from the SSDN underlying assumptions. In particular, the fact that this method assumes a well-defined noise model can be seen as major drawback.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[2] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising. *CoRR*, abs/1811.11127, 2018. 3

[3] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008. 2

[4] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[5] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. Self-supervised denoising. https://github.com/NVlabs/selfsupervised-denoising, 2019. 1

[6] Samuli Laine, Jaakko Lehtinen, and Timo Aila. Self-supervised deep image denoising. *CoRR*, abs/1901.10277, 2019. 1

[7] Marc Lebrun. An Analysis and Implementation of the BM3D Image Denoising Method. *Image Processing On Line*, 2:175–213, 2012. https://doi.org/10.5201/ipol.2012.l-bm3d. 1

[8] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to struc-

tural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 1

[9] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 1