Image Denoising with a Normalizing-Flow Prior

Anonymous CVPR submission

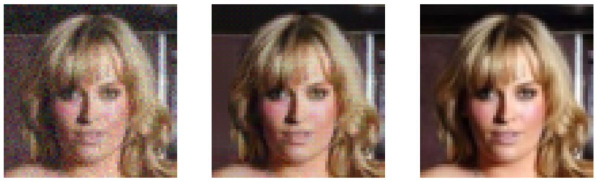
Paper ID *****

Abstract

We introduce a novel method for Gaussian image noise removal based on normalizing flows. A pre-trained model of the image probability density function (PDF) is used as a prior in Bayesian inference. In effect, the method combines noisy observation with prior knowledge transferred from external data to achieve a denoising procedure aware of image content. The model is thus able to produce good results from very corrupted data. Our preliminary results show, that the technique can significantly decrease the difference between observed noisy images and reference images.

1. Introduction

Image denoising is a fundamental task in digital image processing aimed at reducing or eliminating the unwanted noise present in images. Noise can be introduced during image acquisition, transmission, or storage, and it often degrades the visual quality and hinders further analysis or interpretation of the image. The goal of image-denoising techniques is to effectively suppress the noise while preserving the important details and structures in the image. Various approaches have been developed [3] [7] [5], ranging from simple spatial filtering methods to sophisticated algorithms based on statistical models, machine learning, and deep learning, all aimed at enhancing visual clarity and fidelity of noisy images.



Observed *Denoised* *Reference*

Figure 1. Denoising sample result. The figure depicts the observed image with noise on the left, the denoised image in the centre and the original clear image on the right.

In this work, we propose a technique, which utilizes normalizing flows as an image prior distribution for Bayesian image denoising. We explain common assumptions behind image noise modelling, formulate the image denoising problem through the lens of Bayesian inference and propose a solution. We test the proposed approach on the CelebA dataset with normalizing flow model architecture and training routine from [6]. An example is shown in Fig. 1.

2. Our Approach

2.1. Image Noise Model

Although naively, the noise is usually modelled as Gaussian or Poisson-Gaussian, dependent on the true pixel intensity:

$$N \sim \mathcal{N}_\theta(C) \tag{1}$$

where:

- C is a clear image;
- N is an image with noise;
- $\mathcal{N}$ is the noise distribution, e.g. Gaussian;
- θ are noise distribution parameters;

In this work, we assume, that the image noise distribution $\mathcal{N}$ along with θ parameters are known. The goal of the inference will be to find a clear image C that matches an observed N.

2.2. Clear Image Estimation

The posterior distribution of the image C is given by:

$$p(C|N) = \frac{p(N|C)p(C)}{p(N)} \propto p(N|C)p(C) \tag{2}$$

where:

- $p(C|N)$ – PDF of the distribution of the clear image after observing the noisy image;
- $p(N|C)$ – likelihood of the observed image;

- $p(C)$ – prior image distribution;
- $p(N)$ – evidence;

To estimate the latent clear image C from noisy observation N, we maximize the posterior distribution (MAP):

$$\begin{aligned}\hat{C}_{\text{MAP}} &:= \underset{C\in\Omega}{\operatorname{argmax}} \frac{p(N|C)p(C)}{p(N)} = \\ &= \underset{C\in\Omega}{\operatorname{argmax}}\, p(N|C)p(C) = \\ &= \underset{C\in\Omega}{\operatorname{argmax}} \left[\log p(N \mid C) + \log p(C)\right]\end{aligned} \tag{3}$$

We assumed, that the distribution of the noise is known and that we can easily compute the log-likelihood part of the utility function. Usually, the distribution of the noise is assumed to be Gaussian or Poisson-Gaussian, which can be transformed easily to Gaussian via the GAT transformation [1]. In such a case, the log-likelihood will be a sum of squares of residuals.

On the other hand, the modelling of the prior image distribution is highly non-trivial and requires sophisticated tools, that allow for flexible modelling of high-dimensional distributions, such as normalizing flows.

2.3. Normalizing Flows

Normalizing flows are a powerful class of generative models that allow for flexible and efficient modelling of complex probability distributions. Unlike traditional generative models that directly specify the distribution, normalizing flows transform a simple base distribution into a more complex distribution by applying a sequence of invertible transformations. These transformations progressively deform the base distribution, allowing the model to capture intricate data patterns and dependencies. By leveraging the invertibility of these transformations, normalizing flows enable efficient computation of likelihoods and efficient sampling from the learned distribution. They have found applications in various domains, including image synthesis, density estimation, and anomaly detection, and have emerged as promising tools in the field of deep generative modelling.

A normalizing flow model is a deep density estimator. Let $\mathrm{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a bijective function, C be a random vector, and $Z = \mathrm{f}(C)$. By the change of variables formula:

$$p(C) = p(Z) \left|\det \frac{dZ}{dC}\right| \tag{4}$$

where $\frac{dZ}{dC}$ is the Jacobian matrix of f evaluated at C. In normalizing flows, we assume Z follows a chosen tractable distribution, e.g., standard multivariate Gaussian distribution, and approximate the f function with an invertible neural network trained via maximum likelihood estimation on a dataset of iid samples of C.

There are various deep neural network architectures that allow for modelling bijective functions [6] [4] [2]. They often aim to be provably and easily invertible, as well as make it easy to calculate the log-absolute-determinant of the transformation.

2.4. The Method

Firstly, we train a normalizing flow model to estimate the probability density function of a distribution of clear images. For this task, we need a large dataset of images without noise.

Given an image with noise, we maximize the sum of log-likelihood and log-prior image distribution given in (3) calculated through the pullback measure given in (4). We use an estimate of the clear-image distribution as our prior distribution. The distribution of the image noise is assumed to be known. We use Adam to find the denoised image.

Due to the deficiency of our normalizing flow, we found it is best to constrain the difference between a noisy image and a denoised image. We introduce a new hyperparameter β to the maximization task:

$$\hat{C}_{\beta\text{-MAP}} = \underset{C\in\Omega}{\operatorname{argmax}} \left[\beta \log p(N \mid C) + \log p(C)\right] \tag{5}$$

Through the beta parameter, we can control the behaviour of our method, making it possible to obtain solutions more or less similar to the observed image with noise.

3. Results

3.1. Experimental Setup

In this work, we use the architecture proposed in [6]. We use 4 multi-scale architecture blocks [4], each containing 32 1x1 invertible convolution flow steps.

While the method can, in theory, be used for different noise models, in our experiments we use multivariate Gaussian distribution, where noise in different pixels is independent and has a constant predefined variance σ^2:

$$N \sim \mathcal{N}(C, \sigma^2 I) \tag{6}$$

Our test dataset is comprised of the 32 last images of the CelebA dataset [?]. The images were removed from the training dataset during the first stage of the pretraining normalizing flow.

We compare the results both quantitatively and qualitatively. The denoising routine was tested for different noise gains and different beta values. We measure the difference between the denoised image and the ground-truth clear image with mean-squared error (MSE), structural similarity (SSIM), and peak signal-to-noise ratio (PSNR) [?]. Lastly, we assess results qualitatively.

3.2. Quantitative Results

Our denoising technique employed in this study was evaluated using three commonly used measures: structural similarity index (SSIM) (Tab. 1), peak signal-to-noise ratio (PSNR) (Tab. 2), and mean squared error (MSE) (Tab. 3). The results obtained from these measures consistently indicated similar outcomes.

In most cases, a higher value for the β parameter is desired. The beta parameter governs the trade-off between denoising strength and image fidelity. When β is greater than 1, the denoising algorithm tends to prioritize weaker noise reduction. However, the choice of the optimal β value may vary depending on the desired balance between noise suppression and preservation of fine details.

In almost all cases in the test dataset, the technique significantly improved the image according to all the aforementioned measures. We will now qualitatively assess the results showing, that the actual result can be even better than already shown.

σ	$\beta = 1$	$\beta = 2$	$\beta = 4$	$\beta = 8$	noisy
.025	$.964 \pm .01$	$\mathbf{.981 \pm .01}$	$.976 \pm .01$	$.963 \pm .01$	$.966 \pm .02$
.05	$.899 \pm .03$	$.932 \pm .03$	$\mathbf{.943 \pm .03}$	$.881 \pm .03$	$.894 \pm .04$
.1	$.689 \pm .06$	-	$\mathbf{.841 \pm .06}$	-	$.723 \pm .06$
.2	$\mathbf{.657 \pm .07}$	-	$.557 \pm .07$	-	$.453 \pm .07$

Table 1. SSIM. Results obtained by our method as measured by the SSIM, for different noise gains σ and different β parameters. The last column shows SSIM obtained by the observed image (with noise).

σ	$\beta = 1$	$\beta = 2$	$\beta = 4$	$\beta = 8$	noisy
.025	$37.4 \pm .67$	$\mathbf{39.4 \pm .58}$	$38.2 \pm .47$	$36.8 \pm .23$	$35.5 \pm .08$
.05	$32.2 \pm .44$	$\mathbf{34.2 \pm .46}$	$34.0 \pm .73$	$30.7 \pm .16$	$29.5 \pm .09$
.1	$25.2 \pm .39$	-	$\mathbf{29.1 \pm 1.01}$	-	$23.5 \pm .12$
.2	$\mathbf{22.9 \pm 1.15}$	-	$20.6 \pm .48$	-	$17.5 \pm .09$

Table 2. PSNR. Results obtained by our method as measured by the PSNR, for different noise gains σ and different β parameters. The last column shows PSNR obtained by the observed image (with noise).

σ	$\beta = 1$	$\beta = 2$	$\beta = 4$	$\beta = 8$	noisy
.025	$.018 \pm .00$	$\mathbf{.012 \pm .00}$	$.015 \pm .00$	$.021 \pm .00$	$.028 \pm .00$
.05	$.061 \pm .01$	$\mathbf{.039 \pm .00}$	$.041 \pm .01$	$.085 \pm .00$	$.112 \pm .00$
.1	$.303 \pm .03$	-	$\mathbf{.128 \pm .03}$	-	$.447 \pm .01$
.2	$\mathbf{.528 \pm .16}$	-	$.868 \pm .10$	-	$1.784 \pm .04$

Table 3. MSE. Results obtained by our method as measured by the MSE (scaled by 100), for different noise gains σ and different β parameters. The last column shows MSE obtained by the observed image (with noise).

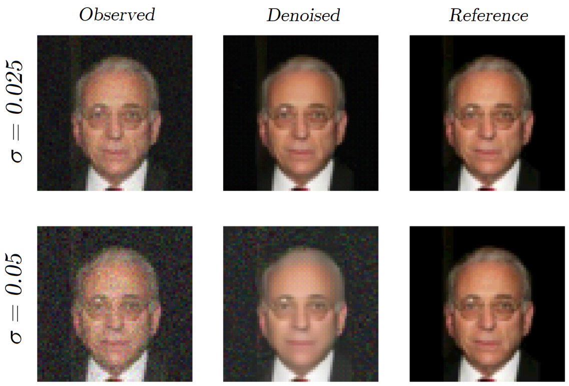


Figure 2. Denoising result for different noise gains. The figure depicts the observed image with noise on the left, the denoised image in the centre and the original clear image on the right.

3.3. Qualitative Results

Consider the example depicted in Fig. 2. For low noise gain the result is very close to the original. For higher noise gain ($\sigma = 0.05$) the Bayes error is high, i.e., there is high uncertainty, in what the image depicts. Because of that, the denoised image differs from the reference image. Our denoising technique improved the quality of the image, however, the hallucinated details do not match the reference image. This example characterises well the behaviour of the method. We reinvent missing details, transferring knowledge from external data, but we are constrained by the information contained in the image.

If hallucinations from the definition are the effect of insufficient information about the semantics of an image, the logical way to counter them is to supply extra information about the image content. In future work, the method could theoretically be modified by changing the prior distribution, to a distribution conditioned on a latent person embedding E containing semantic features of depicted person:

$$p(C|N,E) = \frac{p(N|C,E)p(C|E)}{p(N|E)} \propto p(N|C,E)p(C|E) \tag{7}$$

With additional assumption $p(N|C,E) = p(N|C)$ (the noise depends on the semantics only through the clean image), the difference is only in the prior distribution. A natural extension of the method should somehow relate the E embedding to the Z latent variable.

Fig 3 shows the behaviour of the method with respect to different values of the β parameter. As we can see, for the true MAP estimator ($\beta = 1$), the image is generalized too much and the details are missing. This is caused by deficiencies in the prior estimate, that assigns high-density values to generalized images. We suspect, that the problem would diminish for a more robust normalizing flow model.

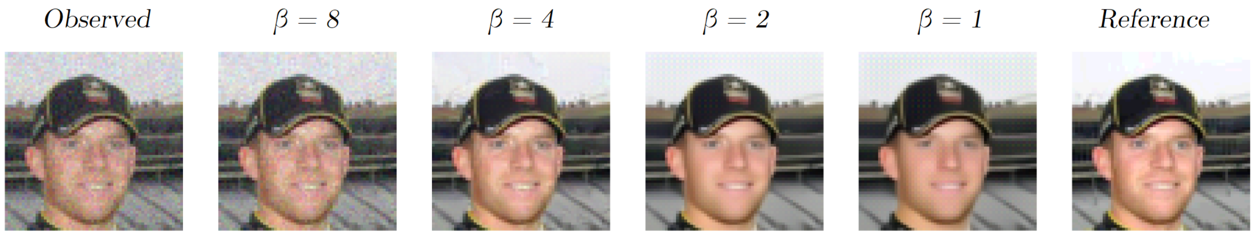


Figure 3. Behaviour of the method for different values of β. The image on the left shows an image with noise. The consecutive four images show denoising results for different values of the β parameter. The higher the β parameter, the closer the resulting image should be to the observed image. The ground-truth reference image is shown on the right for comparison.

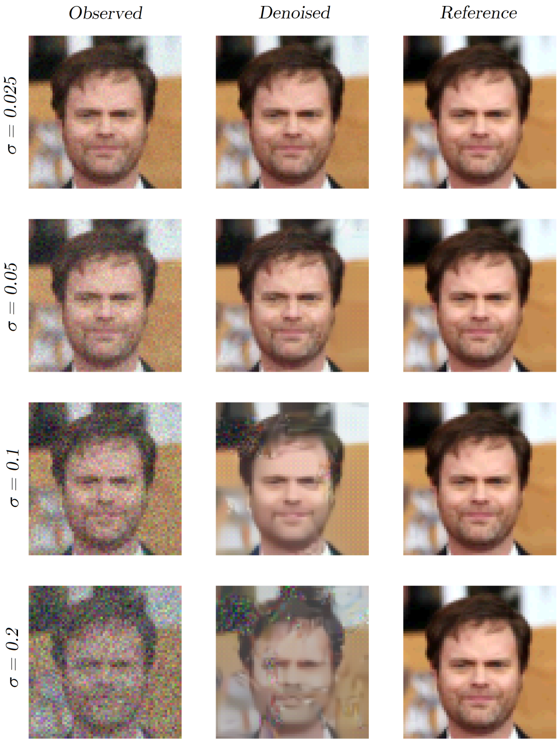


Figure 4. Behaviour of the method for different noise gains σ. The figure depicts the observed image with noise on the left, the denoised image in the centre and the original clear image on the right.

Fig 4 compares the behaviour of the method for different noise gains. Even for the high noise gains, the method tries to reinvent the missing details. However, due to deficiencies of the prior, the results are deformed. Artefacts are visible in the denoised images. Noise is also removed unevenly, as some parts of the images contain more of it, others less.

4. Discussion

While the method's knowledge transfer capability and ability to improve visual quality are advantageous, its inefficiency, resulting in longer processing times, lack of mathematical accuracy due to deficiencies of the prior estimate, and potential semantic alterations pose limitations. Further research is needed to address these concerns and enhance the method's performance and reliability.

5. Conclusion

In this work, we presented a method for image denoising based on Bayesian inference combined with normalizing flows. We showed that the Bayesian framework enables image noise removal, improving images as measured by MSE, PSNR and SSIM. The experiment also shows, that normalizing flows enable the Bayesian framework on complex data, which was previously unavailable. In future work, the efficiency and robustness of the method could be improved. The issue of hallucinations and its possible solutions are to be examined.

References

[1] F. J. Anscombe. The transformation of poisson, binomial and negative-binomial data., 1948. 2

[2] Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. *CoRR*, abs/1811.00995, 2018. 2

[3] Jaeseok Byun, Sungmin Cha, and Taesup Moon. Fbi-denoiser: Fast blind image denoiser for poisson-gaussian noise, 2021. 1

[4] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015. 2

[5] Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1):1–12, 2019. 1

[6] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions, 2018. 1, 2

[7] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. *Advances in neural information processing systems*, 25, 2012. 1