POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Diffusion Models for Image Motion Blur Removal

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Lorenzo Innocenti

Advisor: Prof. Giacomo Boracchi

Co-advisor: Diego Stucchi

Academic year: 2022-23

1. Introduction

Image deblurring consists in restoring an image that is affected by motion blur or that is out of focus. Removal of blur is key in many applications, such as astronomy, microscopy, and digital photography. Neural Networks (NNs) have proven effective in many image-to-image translation problems, and since deblurring can be framed as one of those, studies have applied deep learning methods as a possible solution.
One of the most important elements in deep learning is the dataset employed to train the NN. In image deblurring, the generation of the dataset is not as trivial as, for example, generating a noisy or low resolution dataset.
In this thesis address the current scarcity of deblurring datasets, by proposing a novel image degradation pipeline, and releasing it to the public at `github.com/lorenzoinnocenti/csb-dataset-generator`. Additionally, for the first time, we introduce the application of diffusion models, a novel NN model, to the domain

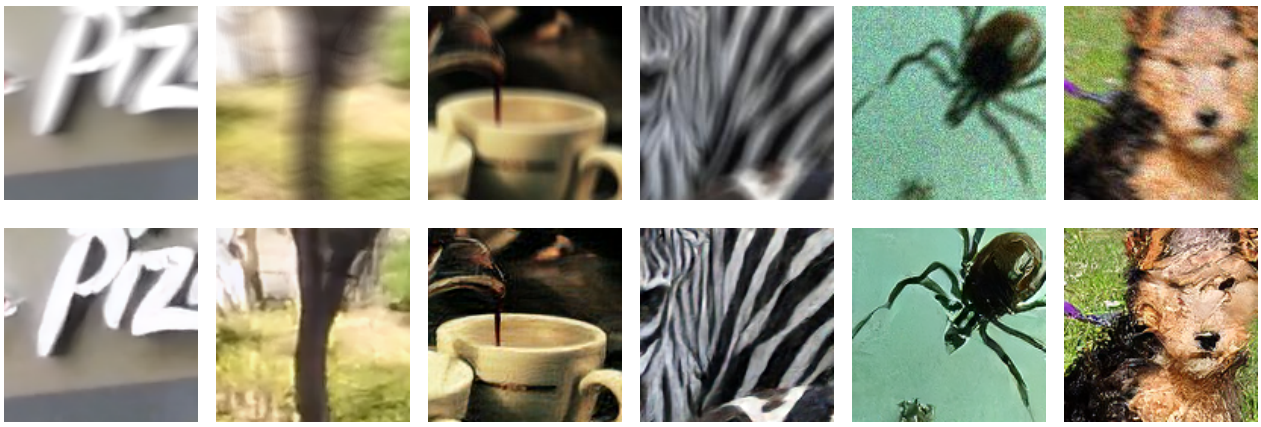

Figure 1: Examples of deblurring using our solution. Input images on the first row and restored images on the second one. First two images taken from the GoPro dataset, then four generated by our synthetic camera shake blur pipeline, the last two of which include the application of noise.

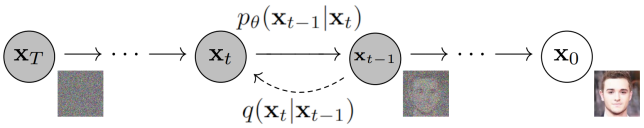

Figure 2: Graphical representation of the Markov chain model considered in [1].

of image deblurring. Figure 1 shows both the images generated through our pipeline and the corresponding restorations by our model.

2. Problem formulation

2.1. Blurring

We denote the blurring degradation process as a function of the sharp image $\mathbf{x}$

$$\mathbf{y} = \Phi(\mathbf{x}), \tag{1}$$

where $\mathbf{y}$ is the blurred image, and Φ is the blurring process. Under the assumption of spatially invariant blur, the degradation can be represented as a convolution with a Point Spread Function (PSF) kernel $\mathbf{k}$

$$\mathbf{y} = \mathbf{k} * \mathbf{x}. \tag{2}$$

Multiple types of blur exist. *Out-of-focus* blur happens when the camera fails to focus the scene onto the sensor. It can be mimicked using a disk-shaped kernel.
Camera shake blur occurs when the camera is in motion while capturing an image. This blur effect can be modeled using a kernel that represents the path of the camera movement during the exposure. This kernel is not as trivial to synthesize as a disk kernel, as it can take into account complex motion patterns, depending on the desired realism of the blur.
In this thesis we follow the degradation model introduced in [2]. First, a trajectory is generated by simulating the motion of a particle in a 2D domain. The particle has an initial velocity and, at each iteration of the generation algorithm, is affected by a random perturbation and by a centripetal component. In addition, with small probability, a random inversion of direction can happen. This trajectory is then sampled in a 2D grid, generating the PSF kernel matrix, in a portion equal to a parameter T, to simulate the exposure time.

Additionally, we apply Poisson noise, to simulate the statistical nature of photon detection, and Gaussian noise, for the amplification of the electrical signal:

$$\mathbf{y} = (\mathbf{u} + \mathbf{n})/T, \tag{3}$$

$$\mathbf{u} \sim \mathcal{P}(\lambda(\mathbf{k} * \mathbf{x})), \quad \mathbf{n} \sim \mathcal{N}(0, \sigma^2), \tag{4}$$

where σ quantifies the thermal and electrical noise of the system, and λ the quantum efficiency of the sensor.
The sum of pixel values in the PSF kernel is equal to T. In [2], the authors utilize this value to simulate the tradeoff between blur and noise, same effect observed in cameras. A smaller value of T reduces the magnitude of blur while amplifying the noise effect, by shrinking the signal range. The multiplication by $1/T$ in equation (3) acts as an amplification factor to restore the full dynamic range of the image. The effect is depicted in Figure 3.
Object motion blur happens when an object moves during the exposure process. This is the most complex type of blur, as it is spatially variant, and cannot be modeled as a convolution. Real-life blurred pictures can have multiple types of blur mixed together.

2.2. Deblurring

Deblurring consists of estimating the sharp image $\tilde{\mathbf{x}}$ by inverting the function Φ:

$$\tilde{\mathbf{x}} = \Phi^{-1}(\mathbf{y}). \tag{5}$$

The deblurring algorithm depends on the blur type: if there is a model for the blur, and the blurring parameters are known, it is possible

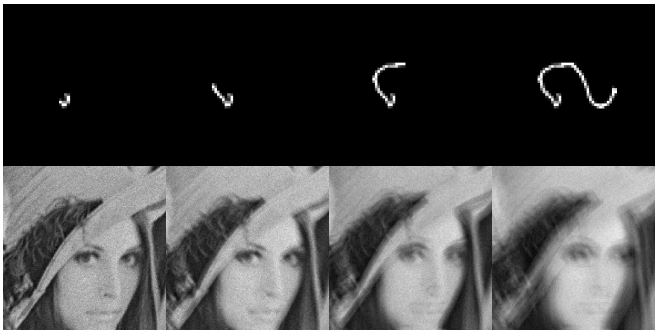

Figure 3: Illustration of the noise-blur tradeoff. Images degraded with the degradation model from [2], using the same trajectory, with $T = 1/8, 1/4, 1/2, 1$.

to use a non-blind approach. Otherwise, if the model is not known, or if it is known but the parameters are not, a blind approach is necessary. In this thesis, we address the problem of blind deblurring.

3. Related work

3.1. Datasets

In image deblurring, the datasets commonly used to train NNs are of two kinds. When a degradation model is employed, the dataset is used as a source of sharp images, which are then synthetically blurred. The most commonly used is ImageNet [3]. To train NNs without any assumptions on the degradation model, datasets like the GoPro set [4] are used. This dataset is obtained by taking a sequence of pictures with an high speed camera. Multiple pictures are averaged to generate a blurred picture, and the middle one is taken as sharp version. It is the most used in this field, and it provides a way to compare the performance with other studies in literature. It contains both camera shake and object motion blur, but lacks in image diversity, as all the pictureas are taken in similar conditions and by the same camera.

3.2. Deblurring with noise and blur

In Section 2.1 we presented the degradation model for camera shake blur from [2]. The authors test the performance of the reconstruction of various image deconvolution algorithms for deblurring, at varying T values. By doing so, they show that, in case of linear blur, the reconstruction presents a clear optimal performance at a T value that depends on the PSF and the noise levels. They also show that, in case of camera shake blur, no clear optimal value can be found, and that the reconstruction performance levels off after a certain T value.

3.3. Diffusion models

Diffusion Models (DMs) [1] are a recently developed generative model. As for all generative models, their goal is to learn the probability distribution of a training set so that we can use it to generate new data with the desired features. Compared to GANs, DMs offer a more stable training result, and often a higher performance [5], at the drowback of longer training and inference times.

In DMs, the generative process is modeled as navigation on a Markov chain, as shown in Figure 2. The process known as *forward diffusion* (q) begins with an image sampled from a real image distribution and gradually introduces incremental amounts of Gaussian noise, until the image can be regarded as entirely composed of noise. The aim of the DMs is to learn the inverse process, referred to as *reverse diffusion* (p_θ). To do so, we train a NN to predict the noise at each step, from the partially noisy image and the index of the step of the diffusion process. After the training process, the DM takes as input a seed in the form of Gaussian noise and navigates the Markov chain to produce an image, as the result of the iterative noise removal.

The architecture in [1] is a residual U-Net with skip connections. The model incorporates global attention blocks [6] to enhance expression power and receptive field. The NN parameters are shared among the indexes of the Markov chain, so a positional encoding [6] block is used to inject the index value into the noise estimation. DMs can be conditioned on an input image, and used as an image-to-image translation NN. To do so, the input noise seed, and all the partially noisy images, are concatenated with the input image. This architecture is called conditional DM.

4. Contributions

4.1. Synthetic camera shake dataset

We present a novel dataset generation pipeline, which adapts the camera shake blur model from Section 3.2 for the training of NNs. The pipeline can be used to generate datasets composed of images degraded with:

- always the same kernel, to train NNs to deconvolve an image from a particular kernel;
- randomly generated kernels, within predefined motion parameters, to train deblurring NNs;
- random kernels and noise levels, within predefined values, to train deblurring and denoising NNs;
- same trajectory and noise levels, but with varying values of T, to investigate the tradeoff between noise and blur.

We are among the first to propose a randomized camera shake blur pipeline for NN training, and

the first to consider noise in the formulation.

4.2. DMs for image deblurring

In this study, we apply conditional DMs to the problem of image deblurring. Modifications has been proposed to the architecture, which has been shown to improve the generative performance of DMs. We explore some of them to test if they also improve deblurring performance. These modifications include the use of multi head and multi resolution attention [5], BigGAN-style residual blocks [7] and increases of the dimension of the latent representation.
As a mean to overcome the limitation of fixed resolution, imposed by some blocks in the architecture, we implement a technique to deblur arbitrary sized images. We do so by extracting multiple overlapping patches, processing them with our model, and combining them back in the full deblurred picture, by weighting the contribution of each patch with a Hann window. We train and test using the GoPro dataset, to show that DMs are a promising alternative to other deep learning approaches to deblurring.
We employ the camera shake blur pipelines to train DMs for camera shake blur removal, with and without noise. We test on a set without noise to show that our method is more effective than the non-blind Wiener deconvolution at deblurring, and we also show that the model performs similarly on noisy datasets. We also employ the pipeline to show that the performance of our method follows the same behaviour presented in [2].

5. Experiments

5.1. DMs on spatially variant blur

We implement the proposed architecture, and apply the modifications individually, to evaluate their impact on performance. We call the model in [1] *base model*, and the one with all the proposed improvements *improved model*.
We conduct the architecture investigation on the GoPro set. We train on patches randomly cropped from the train dataset, at a resolution of 64×64, to keep a low computational cost. We halve the resolution of the images before cropping, to avoid extracting patches with insufficient details for proper restoration, which is often the case at this resolution. We call this dataset, from now on, *halved GoPro dataset*. We test on the halved GoPro test set, by extracting the central 64×64 pixels patch.
Inference time is usually long with DMs, so early stopping is not used in this setting. We train each model for 4000 epochs of 1024 samples. This number is obtained via an analysis on convergeance done on a validation set, split from the training set, and is a compromise between performance and training time. Results are gathered and shown in Table 1. Increasing the latent representation improves results, particularly in terms of LPIPS, but doubles the training and inference time. BigGAN blocks, multi head and multi resolution attention also improve results, with little time increase. Combining all the modifications gives the best performance.
We compare our improved model against some NN-based blind deblurring methods. We train a model at a resolution of 128×128, for 4000 epochs of 1024 patches, on the full resolution GoPro train set. The results of the tests are in Table 2. *Patches* refers to the performance on the central patches of all the images in the test set. *Full res* refers to the performance of the improved model on full resolution images, obtained with the arbitrary resolution algorithm. Due to time constraints, we test on a subset of 110 images. As we use a subset of the test set, we cannot directly compare the metrics, but they nontheless provide the suggestion that DMs are a valid alternative to GANs for image deblurring. We expect this method to outperform the alternatives if trained on larger patches, for more time, and with proper image augmentation.

	PSNR ↑	SSIM ↑	LPIPS ↓
Base	27.10	0.814	0.158
Increased repr.	28.66	0.860	0.116
Multi head att.	27.61	0.829	0.146
Multi res att.	27.54	0.825	0.147
BigGAN block	27.81	0.838	0.146
Improved	**29.58**	**0.880**	**0.102**

Table 1: Results from the experiments in Section 5.1, on the halved GoPro test set.

5.2. DMs on camera shake datasets

In this section, we employ our implementation of the degradation model from [2] to train DMs. In this section we use the improved model at a

	PSNR ↑	SSIM ↑	LPIPS ↓
Ours (patches)	28.30	-	-
Ours (full res)	28.00	0.870	0.183
Sun et al. [8]	24.64	0.842	-
DeblurGAN [9]	27.20	0.954	-
DeepDeblur [4]	28.30	0.917	0.182
SRN [10]	30.10	0.932	0.788
DeblurGAN v2	29.55	0.934	0.253

Table 2: Comparison with other blind deblurring, NN baseline approaches, tested on the GoPro set. The missing values are due to the limited use of LPIPS in literature.

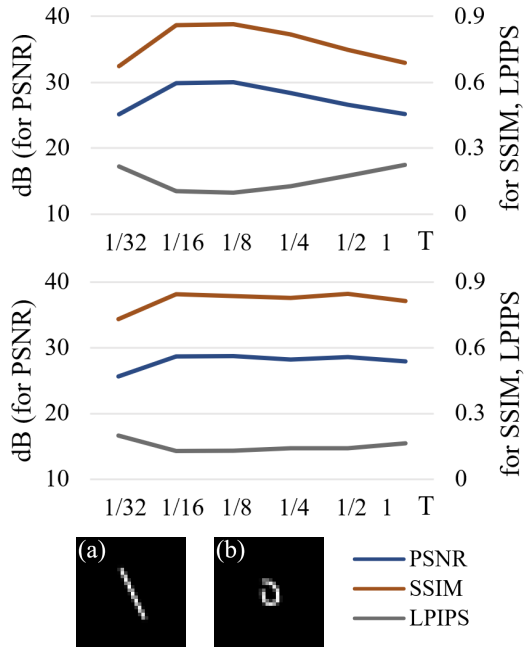

Figure 4: Performance of the reconstruction on constant trajectory, as exposure time changes. Top diagram refers to (a), bottom refers to (b).

resolution of 64×64, and train for 4000 epochs of 1024 samples each. We test the image deblurring performance on three different magnitudes of blur, against the performance of the Wiener deconvolution algorithm, as baseline. We train three models on different dataset settings, with three dimensions of PSF kernels. We use randomly generated trajectories, with maximum length of half of the PSF kernel size. We apply the degradation to randomly extracted patches from the ImageNet training dataset. We avoid the application of noise for this test. We store 1024 test images for each training set, generated by degrading images from the ImageNet test set with the same settings, along with the kernels

		PSNR ↑	SSIM ↑	LPIPS ↓
16p	Blurred	27.47	0.752	0.202
	Wiener	29.52	0.870	0.144
	Ours	32.83	0.917	0.058
32p	Blurred	27.47	0.752	0.202
	Wiener	25.53	0.737	0.259
	Ours	28.83	0.796	0.135
64p	Blurred	25.24	0.652	0.283
	Wiener	22.64	0.586	0.406
	Ours	25.76	0.702	0.176

Table 3: Results on deblurring on datasets without noise, at different sizes of blurring kernels. PSF dimensions are reported in the first column.

		PSNR ↑	SSIM ↑	LPIPS ↓
no noise	In	27.47	0.752	0.202
	Out	28.83	0.796	0.135
$\sigma = 1$,	In	24.66	0.643	0.281
$\lambda = 768000$	Out	25.67	0.710	0.184
$\sigma = 2$,	In	22.40	0.481	0.336
$\lambda = 192000$	Out	23.43	0.613	0.248
$\sigma = 4$,	In	19.32	0.312	0.438
$\lambda = 48000$	Out	22.69	0.572	0.309

Table 4: Results at different levels of noise. PSF dimension of 32p, noise levels in the first column. In the *In* row the degraded metrics, and in the *Out* row the ones restored by our solution.

used for degrading them, to be used to compare the results with the Wiener method.
To test the Wiener performance, we tune the regularization parameter on the training sets, and we use the best performing regularization value on the test sets. We train three improved DMs on the three datasets, and we report the performance metrics in Table 3. As we can see, the DMs perform better than the baseline algorithm on all the tested conditions, especially in case of large kernels, where Wiener struggles due to the ringing artifacts.
We repeat the same test setup on three other datasets, this time generated by keeping the kernel size fixed but varying the noise levels. We train three models on datasets genetated with increasing levels of noise, and test on test sets of images degraded accordingly, as done previously. We use realistic noise levels, taken from [2]. We report the results in Table 4, where it is shown that our method has comparable per-

formance at the different noise levels as well as on the dataset without noise, suggesting that our method is succesful in both deblurring and noise suppression.
We replicate the experimental setup described in [2]. The experiment consists in training models using datasets containing images from the ImageNet training set, degraded with a fixed trajectory and noise levels, with T varying between 0 and 1. We use the same noise levels as [2]: $\sigma = 0$ and $\lambda = 765000$. To evaluate the reconstruction performance, we utilize six test sets containing the same 100 images from the ImageNet test set, degraded with the same trajectory, noise levels, and T value. This process is repeated for two distinct trajectories, and the results are depicted in Figure 4. The findings validate the observations made on the deconvolution algorithms discussed in [2]: when the trajectory is linear, there is a discernible optimal T; whereas for more intricate trajectories, performance tends to plateau after a certain T value. Additionally, we conducted the experiment using two different trajectories and higher levels of noise, obtaining similar results.

6. Conclusions

This study presented an application of conditional DMs in image deblurring, showing the potential of this approach, and what architecture is the best among the ones analyzed. Our low-resolutions tests with the GoPro dataset achieved promising results that matched other baseline deblurring NNs. We expect that these findings might generalize well to models with an higher resolution, but further research is needed. We presented a novel image degradation pipeline, which incorporates camera shake blur and a realistic noise component. We employed it to show that our method is effective in both deblurring and noise suppression.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[2] Giacomo Boracchi and Alessandro Foi. Modeling the performance of image restoration from motion blur. *IEEE Transactions on Image Processing*, 21(8):3502–3517, 2012.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[8] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 769–777, 2015.

[9] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018.

[10] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018.