Generic Image Restoration with Flow Based Priors

Leonhard Helminger[1,*] Michael Bernasconi[1,*] Abdelaziz Djelouah[2]
Markus Gross[1] Christopher Schroers[2]

[1]Department of Computer Science
ETH Zurich, Switzerland

[2]DisneyResearch|Studios
Zurich, Switzerland

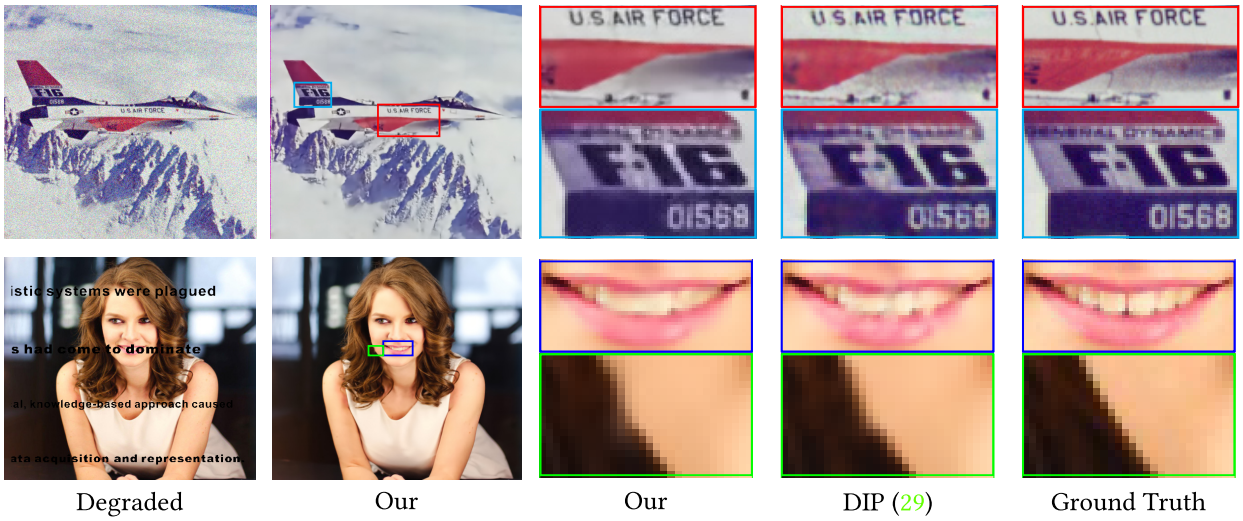

Figure 1: Comparative results with Deep Image Prior (29) on different image restoration tasks. The first example corresponds to denoising whereas the second is image inpainting. Our approach is able to remove the degradation and produces visually more pleasing results in some regions like the text (on the plane) and the mouth.

Abstract

Image restoration has seen great progress in the last years thanks to the advances in deep neural networks. Most of these existing techniques are trained using full supervision with suitable image pairs to tackle a specific degradation. However, in a generic setting with unknown degradations this is not possible and a good prior remains crucial. Recently, neural network based approaches have been proposed to model such priors by leveraging either denoising autoencoders or the implicit regularization captured by the neural network structure itself. In contrast to this, we propose using normalizing flows to model the distribution of the target content and to use this as a prior in a maximum a posteriori (MAP) formulation. By expressing the MAP optimization process in the latent space through the learned bijective mapping, we are able to obtain solutions through gradient descent. To the best of our knowledge, this is the first work that explores normalizing flows as prior in generic image enhancement problems. Furthermore, we present experimental results for a number of different degradations on data sets varying in complexity and show competitive results when comparing with the deep image prior approach.

1. Introduction

In today's digitized world, there is an increased demand to process existing older content. Examples are the archival of photo prints (18) for more reliable long-term data storage, preparing heritage footage (1) for more engaging documen-

* Authors contributed equally to the work

taries, and making classic films and existing catalog contents available to large new audiences through streaming services. This old content is however often in low quality and may be deteriorated in complex ways, which creates a need for *generic* image restoration methods that are able to address a wide range of possibly combined degradations. Image restoration can be formulated as solving the following energy minimization problem:

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \left[\mathcal{L}_{\text{data}}\left(\hat{\mathbf{x}},\ \mathbf{x}\right) + \mathcal{L}_{\text{reg}}(\mathbf{x})\right] \ , \tag{1}$$

where $\hat{\mathbf{x}}$ is the observed image and $\mathbf{x}^\star$ the restored image to be estimated. The first term, $\mathcal{L}_{\text{data}}$, is a data fidelity term which can be problem dependent and ensures that the solution agrees with the observation; the second term, $\mathcal{L}_{\text{reg}}(\mathbf{x})$, is a regularizer that typically encodes certain smoothness assumptions on the expected solution and thus pushes it to lie within a given space. From a Bayesian viewpoint, the posterior distribution of the restored image is $p(\mathbf{x}|\hat{\mathbf{x}}) \propto p(\hat{\mathbf{x}}|\mathbf{x})p(\mathbf{x})$. This allows rewriting the above restoration problem into the following equivalent maximum a posteriori (MAP) estimate:

$$\mathbf{x}^\star = \arg\max_{\mathbf{x}} \ \log(p(\mathbf{x}|\hat{\mathbf{x}})) \tag{2}$$

$$= \arg\max_{\mathbf{x}} \ \underbrace{log\left(p\left(\hat{\mathbf{x}}|\mathbf{x}\right)\right)}_{\text{data}} + \underbrace{\log p(\mathbf{x})}_{\text{regularizer}} \ , \tag{3}$$

which makes it more explicit that the regularizer should model prior knowledge about the unknown solution. Many handcrafted priors have been proposed reflecting desired properties based on total variation (28), gradient sparsity (11) or the dark pixel prior (13). More recently, learning based priors have been explored, in particular the usage of denoising autoencoders (DAEs) as regularizers for inverse imaging problems (21). Building on DAEs, Bigdeli *et al.* (6) propose to use a Gaussian smoothed natural image distribution as prior. In a different direction, Ulyanov *et al.* (29) showed that an important part of the image statistics is captured by the structure of a convolutional image generator even independent of any learning.

All existing methods proposed alternatives and approximations to the true image prior $p(\mathbf{x})$ in Equation 2. However, with deep normalizing flows, we have an approach for a tractable *and* exact log-likelihood computation (10). Therefore, we propose to use normalizing flows for capturing the distribution of target high quality content to serve as a prior in the MAP formulation. In addition to this, the inference of the latent value that corresponds to a data point can be done exactly without any approximation since our generative model is invertible. We use this learned bijective mapping to express the MAP optimization process in the latent space and are able to obtain solutions through gradient descent. Concurrent to our work, Asim *et al.* (4) also explored using invertible neural networks priors for inverse problems. However we demonstrate good results on a more diverse set of images and at arbitrary resolution, while only face images were used in (4). This is thanks to our proposed additional losses to improve the data manifold of the base distribution (latent space) for the MAP optimization. In a number of experiments, we explore our approach for different degradations on data sets of varying complexity and we show that we can achieve competitive results as illustrated in Figure 1.

The contribution of this paper is three fold: 1) to the best of our knowledge, our work is the first using normalizing flows to learn a prior for generic image restoration; 2) we take advantage of the bijective mapping learned by our model to express the MAP problem of image reconstruction in latent space, where gradient descent can be used to estimate the solution; 3) we propose using new loss terms during model training for regularizing the base distribution space which yields a better behavior during the MAP inference.

Could this be removed: Our paper is organized as follows. In Section 2, we recap important background regarding normalizing flow before describing our method in Section 3. Section 4 covers important related work and Section 5 discusses our experimental results. We give our conclusions in Section 6.

2. Normalizing Flow

Borrowing the notation from Papamakarios *et al.* (22), let's consider two random variables X and U that are related through the reversible transformation $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$, $\mathbf{x} = T\left(\mathbf{u}\right)$. In this case, the distribution of the two variables are related as follows:

$$p_X(\mathbf{x}) = p_U\left(\mathbf{u}\right)\left|\det J_T\left(\mathbf{u}\right)\right|^{-1} \ , \tag{4}$$

where $\mathbf{u} = T^{-1}\left(\mathbf{x}\right)$ and $J_T\left(\mathbf{u}\right)$ is the Jacobian of T. Here, the determinant preserves total probability and can be understood as the *amount* of squeezing and stretching of the space induced by the transformer T. The objective of normalizing flows (26) is to map a base distribution to an arbitrary distribution through a change of variable. In practice, a series $T_1, \ldots, T_K$ of such mappings are applied to transform the base distribution into a more complex multi-modal one

$$\mathbf{x} \overset{T_K^{-1}}{\longleftrightarrow} \mathbf{h}_{K-1} \overset{T_{K-1}^{-1}}{\longleftrightarrow} \mathbf{h}_{K-2} \cdots \mathbf{h}_1 \overset{T_1^{-1}}{\longleftrightarrow} \mathbf{u} \ , \tag{5}$$

$$p_X\left(\mathbf{x}\right) = p_U\left(T^{-1}\left(\mathbf{x}\right)\right) \prod_{k=1}^{K} \left|\det \frac{d\mathbf{h}_{k-1}}{d\mathbf{h}_k}\right| \ , \tag{6}$$

where we define $\mathbf{h}_K \triangleq \mathbf{x}$ and $\mathbf{h}_0 \triangleq \mathbf{u}$. It is clear that computing the determinant of these Jacobian matrices, as well as the function inverses, must remain easy to allow their integration as part of a neural network. This is not the case for arbitrary Jacobians and recent successes in normalizing

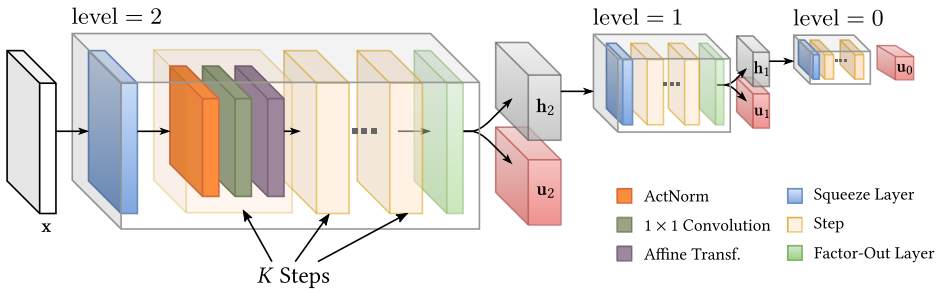

Figure 2: Overview of the normalizing flow architecture. The input image $\mathbf{x}$ is processed by an $L = 3$ level network, where each level consists of a squeeze operation followed by a series of K steps. Each step is a succession of *ActNorm*, 1×1 *convolution* and an *affine layer*. The image latent representation is $(\mathbf{u_0}, \mathbf{u_1}, \mathbf{u_2})$. The number of levels and steps can be adapted to the complexity of the data.

flow are due to the proposition of invertible transformations with easy to compute determinants.

Normalizing flows as generative model. Recent works have shown the great potential of using normalizing flow as generative model (16; 10) where an image observation $\mathbf{x}$ is generated from a latent representation $\mathbf{u}$

$$\mathbf{x} = T_\theta(\mathbf{u}) \qquad \text{with} \qquad \mathbf{u} \sim p(\mathbf{u}) \ . \tag{7}$$

Here $\mathbf{x} \in \mathcal{X}$ is a high-dimensional vector, T_θ denotes a composition of invertible transformations, and $p(\mathbf{u})$ is the base distribution e.g. a normal distribution. Considering a discrete set $\mathcal{D}$ of N natural images, the flow based model learns the parameterized distribution, $p_\theta(\mathbf{x})$, by minimizing the following log-likelihood objective:

$$\mathcal{L}_{nll}(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} -\log p_\theta\left(\mathbf{x}^{(i)}\right). \tag{8}$$

where $\mathbf{x}^{(i)}$ are the images in the training dataset. In the next section, we will describe our approach for leveraging flow based models for various image restoration applications.

3. Generic Restoration with Flow Based Priors

By training a generative flow model as described in the previous section, we learn a mapping T_θ from a latent space $\mathcal{U}$, with a known base distribution $p(\mathbf{u})$, to the complex image space $\mathcal{X}$. In this work, we propose to use the capacity of normalizing flows to compute the exact likelihood of images $p_\theta(\mathbf{x})$, as prior in the image restoration problem

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \underbrace{-\log p(\hat{\mathbf{x}}|\mathbf{x})}_{\text{data}} \underbrace{-\log p_\theta(\mathbf{x})}_{\text{prior}} . \tag{9}$$

In addition to the prior, we also take advantage of the bijective mapping in normalizing flows to rewrite the optimization with respect to the latent $\mathbf{u}$

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} [\underbrace{-\log p(\hat{\mathbf{x}}|\, T_\theta(\mathbf{u}))}_{\text{data}} \underbrace{-\log p_\theta(T_\theta(\mathbf{u}))}_{\text{prior}}] \tag{10}$$

since $\mathbf{x} = T_\theta(\mathbf{u})$. With this new formulation, we are leveraging the learned mapping between the complex input space (the image space $\mathcal{X}$) and the base space (the latent space $\mathcal{U}$) that follows a simpler predefined distribution. This new space has interesting properties where the optimization problem is easier to solve. In particular in this work we solve it through an iterative procedure, similar as during training, where gradient descent is applied on the latents according to

$$\mathbf{u}^{t+1} = \mathbf{u}^t - \eta \nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}; \hat{\mathbf{x}}, \theta) . \tag{11}$$

Here $\mathcal{L}(\mathbf{u}; \hat{\mathbf{x}}, \theta)$ abbreviates the objective defined in equation 10 and η is the weighting applied to the gradient. We used the Adam optimizer (15) to compute the gradient steps.

The model is generic and once trained on *target quality* images, different applications can be considered by adapting the data loss term. In this work we use a generic data fidelity term between the input image $\hat{\mathbf{x}}$ and the restored result $\mathbf{x} = T_\theta(\mathbf{u})$:

$$\mathcal{L}_{\text{data}}(\hat{\mathbf{x}},\ \mathbf{u}) = -\log p(\hat{\mathbf{x}}|\, T_\theta(\mathbf{u})) \tag{12}$$

$$= \mathbf{m} \odot \lambda ||\hat{\mathbf{x}} - T_\theta(\mathbf{u})||_2^2 , \tag{13}$$

where $\odot$ is the Hadamard product. The mask $\mathbf{m}$ is a binary mask that indicates pixel locations with valid color values and allows to handle the inpainting scenario. The parameter λ controls the deviation tolerance from the original degraded input $\hat{\mathbf{x}}$. Next we provide details on the normalizing flow architecture used, the training losses, and our coarse to fine optimization procedure.

3.1. Generative Flow Architecture

The proposed generative model is based on the architecture described by Kingma and Dhariwal (16). We first present the individual building layers

- **Activation normalization.** It performs an affine transformation on the activations using a learned scale and bias parameter per channel (16).
- **Invertible 1×1 convolution.** The random permutation of channels are replaced with this convolution (16).
- **Affine transformation.** This layer is a coupling (9) that splits the input into two partitions, where one is the input for the conditioner, a neural network to modify the channels of the second partition.
- **Factor-out layers.** Factoring-out parts of the base distribution (10) allows a coarse to fine modeling.

Using these layers, we propose the model illustrated in Figure 2. It consists of L levels, each one is a succession of K steps, where a step is defined as the composition of the layers: *ActNorm*, *Invertible* 1×1 *convolution* and *Affine*. At the end of each intermediate level l, the transformed values (*latents*) are split in two parts $\mathbf{h}_l$ and $\mathbf{u}_l$, with the factor-out layer. The parameters (μ_l, σ_l) of the conditional distribution $p(\mathbf{u}_l \mid \mathbf{h}_l)$ are predicted by a neural network. In our case, this is a zero initialized 2D convolution as proposed in (16). In the experimental part and in supplementary material, we provide more details about the architecture used for each dataset.

3.2. Training and Latent Space Regularization

When using normalizing flows to learn a continuous distribution, the input images have to be *dequantized*. Following common practices in generative flows, we redefine the negative log-likelihood objective (nll) of equation 8

$$\mathcal{L}_{nll}(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} -\log p_\theta \left(\mathbf{x}^{(i)} + \epsilon \right) . \tag{14}$$

Here ϵ is uniformly sampled from $[0, 1]$. This model is sufficient for simple datasets as we show in the experimental section with the MNIST examples (see Figure 3). However for more complex data, a regularization of the learned latent space is needed. The main objective is to structure this space in a beneficial way for the optimization.

Latent-Noise loss. In order to enforce some regularization of the latent space, we add uniform noise to the latents $\mathbf{u}_\xi = \mathbf{u} + \xi$ where $\xi \sim \mathcal{U}(-0.5, 0.5)$. The proposed loss term

$$\mathcal{L}_{ln} = ||T_\theta(\mathbf{u}_\xi) - \mathbf{x}||_2^2 \tag{15}$$

penalizes parameters θ that would map back $\mathbf{u}_\xi$ far from the initial input image $\mathbf{x}$. It is interesting to note that this loss does not make any assumption regarding the degraded images, but it still results in a latent space better suited for our optimization problem.

Auto-Encoder loss. If we consider the model illustrated in Figure 2, the image $\mathbf{x}$ is mapped to its representation $(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2)$. From only the latent value $\mathbf{u}_0$, we compute $\tilde{\mathbf{x}}$ by sampling the most likely intermediate values $\tilde{\mathbf{u}}_l \sim p(\mathbf{u}_l \mid \mathbf{h}_l)$. Since we use a Gaussian distribution, this corresponds to the mean value of the predicted distribution. The proposed loss

$$\mathcal{L}_{ae} = ||\tilde{\mathbf{x}} - \mathbf{x}||_2^2 \tag{16}$$

forces the model to store sufficient information in the deepest level to reconstruct the image. This allows a more robust coarse-to-fine strategy during the optimization.

Image-Noise loss (optional). The *Image-Noise*-loss $\mathcal{L}_{in}$ works similarly to the *Latent-Noise*-loss $\mathcal{L}_{ln}$. The difference is that the noise is added to the image $\mathbf{x}$ and distortion is measured on the encoding $\mathbf{u} = T_\theta^{-1}(\mathbf{x})$.

$$\mathcal{L}_{in} = ||T_\theta^{-1}(\mathbf{x}) - T_\theta^{-1}(\mathbf{x} + \eta)||_2^2, \tag{17}$$

where $\eta \sim \mathcal{U}(-10, 10)$. We consider this loss to be optional as we found that it only made the optimization slightly faster when the model was trained with this loss and was only used for tests on the Div2K dataset (3).

The final training loss for the normalizing flows is

$$\mathcal{L} = \mathcal{L}_{nll} + \beta_{ln}\mathcal{L}_{ln} + \beta_{ae}\mathcal{L}_{ae} + \underbrace{\beta_{in}\mathcal{L}_{in}}_{\text{optional}}, \tag{18}$$

where β_{ln}, β_{ae} and β_{in} are the weightings for each loss term. We used $\beta_{ln} = 100$, $\beta_{ae} = 1$ and $\beta_{in} = 100$. The weight values are chosen such that all losses have similar contributions to the final loss. The same values were used for all the datasets. The ablation study in the experimental section shows the necessity of training the generative flow model with the different loss terms.

3.3. Coarse-To-Fine Optimization

The optimization procedure described in Equation 11 is iterative and we need to set its initial value $\mathbf{u}^0$. In order to choose a good starting point, we leverage the introduced multi-scale architecture. Our starting point is

$$\mathbf{u}^0 = (\hat{\mathbf{u}}_0, \tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2) \quad \text{with} \quad \hat{\mathbf{u}}_0 \quad \text{defined by} \quad T_\theta^{-1}(\hat{\mathbf{x}}). \tag{19}$$

The values of the other components, $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{u}}_2$, are sampled as the mean values of the respective predicted distributions, namely $p(\mathbf{u}_1 \mid \mathbf{h}_1)$ and $p(\mathbf{u}_2 \mid \mathbf{h}_2)$. As our auto-encoder loss enforces the possibility to reconstruct the image from $\hat{\mathbf{u}}_0$ only, this lowest level contains coarse image information while details are stored in the upper levels. This is advantageous for image restoration tasks where the degradation often affects the *detail* of an image.

Given this starting point, the optimization is done in a coarse-to-fine fashion. First, only the lowest level variables

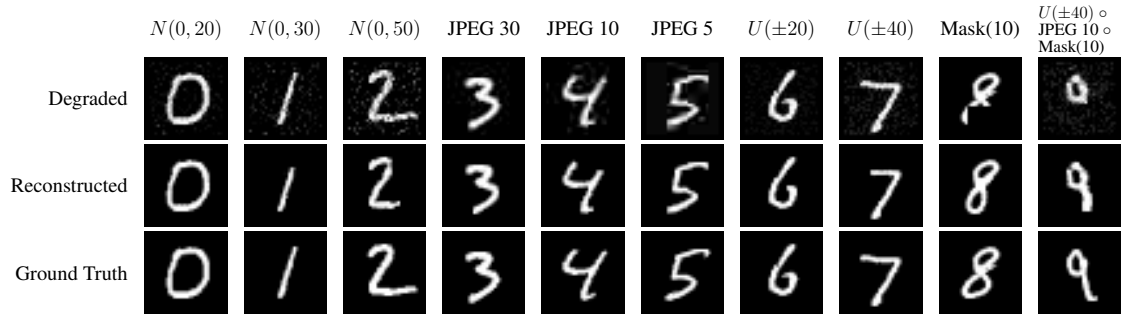

Figure 3: Results produced by a single-level normalizing flow trained on the MNIST dataset. Each column corresponds to a different type of degradation. From top to bottom the degraded image, the reconstructed image and the ground truth are shown. The digits in the first three columns are degraded with random noise $N(0, \sigma)$. JPEG compression with different quality settings is used for digits three to five. Additive uniform noise $U(\pm A)$ was added to digits six and seven. A patch of size 10x10 was masked out of digit eight. Finally, for digit nine, we applied a composition of all.

are optimized while the upper levels are respectively sampled from the predicted means. These are then progressively included in the optimization:

$$\begin{aligned} \mathbf{u}_0^{t+1} &= \mathbf{u}_0^t - \eta \nabla_{\mathbf{u}_0} \mathcal{L}\left(\mathbf{u}; \hat{\mathbf{x}}, \theta\right) \ , \\ (\mathbf{u}_0, \mathbf{u}_1)^{t+1} &= (\mathbf{u}_0, \mathbf{u}_1)^t - \eta \nabla_{(\mathbf{u}_0, \mathbf{u}_1)} \mathcal{L}\left(\mathbf{u}; \hat{\mathbf{x}}, \theta\right) \ , \\ (\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2)^{t+1} &= (\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2)^t - \eta \nabla_{(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2)} \mathcal{L}\left(\mathbf{u}; \hat{\mathbf{x}}, \theta\right) \ . \end{aligned} \tag{20}$$

With this coarse-to-fine scheme, we are able to incrementally refine the reconstructed images by making sure that the lower level information is correct first.

4. Related Work and Discussion

Despite the success of supervised deep learning approaches for dedicated image restoration problems such as super-resolution (30; 33), denoising (31), inpainting (24) or a combination of them (23), one important drawback is the need for retraining whenever the specific degradation or its parameters change. Some recent works (8; 5) have investigated the blind setting for super-resolution. However that concerns the parameters of the degradation only and such solutions are not applicable to an unknown degradation.

When addressing generic image restoration problems, the common approach is to consider the Bayesian perspective where recovering the original image is expressed as solving a maximum a posteriori (MAP) problem. The objective function consists of a fidelity term and a regularization term. The fidelity term can be problem specific and easier to express than the prior that is supposed to reflect desired properties of the reconstructed image. Existing handcrafted priors are based on total variation (28), gradient sparsity (11) or the dark pixel prior (13).

Recently, several works have investigated the usage of CNNs as priors. For example, a deep CNN trained for image denoising can effectively be used as prior in various image restoration tasks (27; 32). Additionally, Meinhardt *et al.* (21) provide new insights on how the denoising strength of the neural network relates to the weight on the data fidelity term. Bigdeli *et al.* (6) define a utility function that includes the smoothed natural image distribution and relate this to denoising autoencoders. In a different direction, Ulyanov *et al.* (29) showed that an important part of the image statistics is already captured by the structure of a convolutional image generator itself, independent of any learning. This work was further analyzed from a Bayesian perspective (7) and combined with a denoising autoencoder prior (20). In (12) the authors propose a method for image segmentation using multiple deep image priors. and show how various image restoration tasks e.g. dehazing or watermark removal can be formulated and solved as such segmentation problems. More recently, Ren *et al.* (25) propose a MAP formulation, to learn the blur kernel and clean image of a blurry picture. They leverage the insights from (29; 12) to regularize the image as well as the kernel. Their approach works well in cases where the degradation can be expressed as a convolution (i.e. blurring).

The idea presented in our work stems from recent developments in normalizing flows (9; 10; 16) and their promising capacity of learning a bijective mapping from a space with a prescribed distribution to the complex space of images, additionally providing exact log-likelihood tractability. Some recent works already explore their usage for image restoration and enhancement problems. For example, Abdelhamed *et al.* (2) use normalizing flows to estimate the distribution of real noise, which can be leveraged to generate training data for denoising. In the case of super-resolution, Lugmayr *et al.* (19) introduce a flow based method for image super resolution, conditioned on the low resolution images. The authors also show how the method can be applied to other types of degradation but our experiments demonstrate

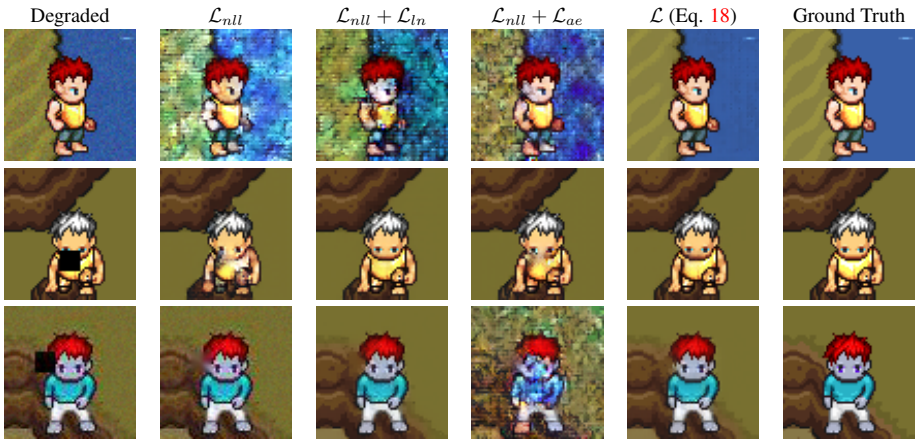

Figure 4: Restoration of degraded Sprites: (top) denoising of Gaussian noise $N(0, 5)$; (middle) inpainting and (bottom) combines denoising; inpainting and JPEG artifact removal. The columns correspond to different normalizing flow models, each one trained with the indicated loss term. Results show the importance of using all the proposed loss terms.

that our approach produces better results.

Using a learned prior that only depends on properties of high quality images is an exciting direction, as this removes the need to rely on other assumptions that are either explicit, in the case of handcrafted solutions, or implicit in the case of denoising autoencoders. Concurrent to our work, Asim *et al.* (4) also explored using invertible neural networks as signal priors for inverse problems such as denoising, compressive sensing, and inpainting. Our solution however goes beyond as we are able to demonstrate good results on a more diverse set of images and at arbitrary resolution. even outperforming DIP (29) in some cases, while only face images were used in (4). This is thanks to our proposed additional losses to improve the image enhancement capabilities.

5. Experiments

In this section we explore the usage of our proposed solution for generic image restoration tasks. We show results on two synthetic datasets, the MNIST and the self generated Sprites, and on real images. We also include comparisons with Deep Image Prior (DIP) (29), Double-DIP (12) and SRFlow (19).

Since we do not focus on a specific degradation during training, our proposed approach can be applied on various types of restoration problems. In this work we present results on three different types of image degradation: noise (uniform and normal), JPEG compression artifacts, and missing regions. We also include the composition of multiple of these degradations. The noisy images are generated by adding i.i.d. samples of noise to the pixel values, with noise distributed according to $\mathcal{U}(\min, \max)$ or $\mathcal{N}(0, \sigma)$. The varying degrees of JPEG artifacts are generated by using different levels (10 to 70) for the JPEG compression. For the inpainting task, we masked multiple regions of size 10×10 pixels. An overview of the used degradations is visualized in Figure 3.

Proof of concept using the MNIST. As a first step we tested our flow based image prior on the well studied MNIST dataset (17). Given the simplicity of this dataset, the model used for this experiment consists of a single-level $L = 1$ with $K = 16$ steps. We choose the base distribution $p(\mathbf{u})$ to be a Gaussian with unit variance and a trainable mean. Further, a ResNet (14) with 2 blocks and $C = 128$ intermediate channels, was used to learn the parameters for the affine transformations. More details about the architecture are provided in supplementary material.

Given a degraded image $\hat{\mathbf{x}}$ the goal is to find the most likely image $\mathbf{x}^\star$ by solving the optimization problem of Equation 10. Given the simplicity of the data set, we use the mean of the base distribution $p(\mathbf{u})$ as starting point $\mathbf{u}^0$. It can be seen in Figure 3 that this is sufficient to enhance the binary digits for any degradation. A related experiment was conducted by Dinh *et al.* (9), where the degraded digits are enhanced by maximizing the probability of the image trough back propagation to the pixel values. This related experiment is equivalent to only considering the prior term in our Equation 10.

Ablation study using the Sprites dataset. Each image in the Sprites dataset consists of a figure performing some pose in front of a random background. Figures are centered in the image and have varying color for hair and clothing. Each image is of size 64×64 (the dataset will be made available upon acceptance). As the images become more complex, it is necessary to use a multi-level architecture and train the normalizing flow model using the loss terms described in Section 3. We increased the capacity of our flow based prior and use $L = 3$ levels, with $K = 8$ steps each. Additional details about the architecture can be found in the supplementary material.

In the optimization, the learning rate η and the data weighting term λ are set to 1 and 99, respectively. The

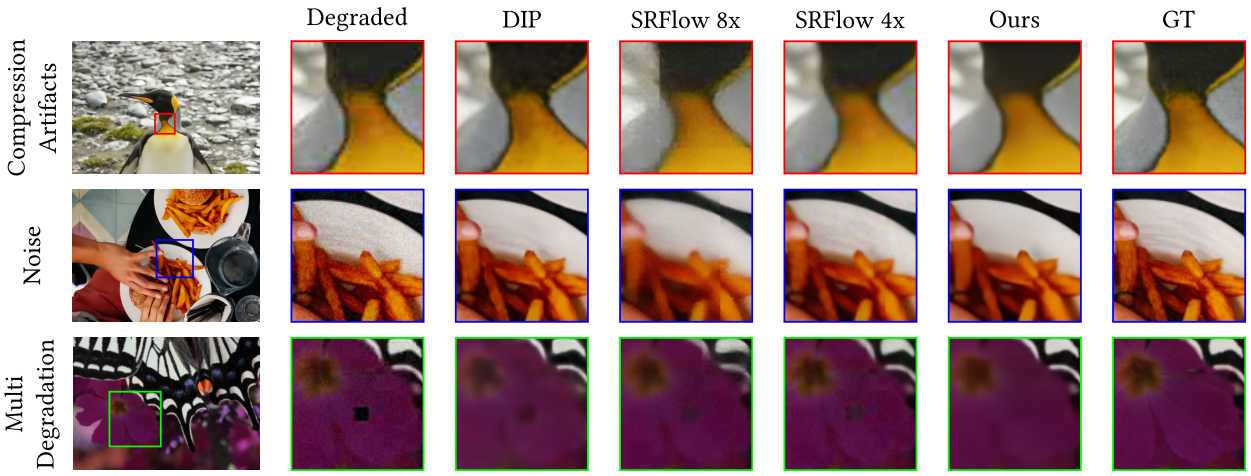

Figure 5: Results on DIV2K dataset. The proposed prior is used to restore images of arbitrary size. Degradations include: (top) JPEG compression artifacts; (middle) denoising; (bottom) and a combination of masked regions, noise and compression artifacts .

gradient descent is done in a coarse to fine way (see section 3.3), each time with 50 update steps per level before including the next one. When all latent levels are included, an additional 150 optimization steps are performed. Using a modern GPU (GeForce GTX 1080 Ti) this amounts to approximately $50s$ processing time per image.

Figure 4 shows image restoration results on this dataset: The first row corresponds to a denoising task, the second is image inpainting and the last combines both in addition to compression artifact removal. Note that these images were not observed during training. Our proposed algorithm (last column) is able to successfully predict missing parts alongside denoising and compression artifact removal.

In addition to our proposed model, we trained variants with different loss combinations to demonstrate the importance of the regularization losses proposed in Section 3.2. Using the negative-log-likelihood loss ($\mathcal{L}_{nll}$) is clearly not sufficient, and a prior trained only with this term is not suited for the latent space optimization.

The most important improvement comes from using the latent-noise loss ($\mathcal{L}_{ln}$). This regularization enforces neighboring elements in latent space to be mapped back to similar images. This is highly beneficial to the gradient descent procedure in latent space and a prior trained with this loss already leads to some good restoration results. The motivation behind this loss term comes for the observation that the image optimization had a tendency to diverge suddenly and irrecoverably. This could be explained by ill behaved regions in the latent space where gradients explode. To alleviate this problem we want to ensure that similar latent space representations correspond to similar images. The idea here being that two similar images probably also have similar likelihoods which would results in small gradients. The Latent-Noise loss encourages this, as it penalizes similar latent space representations resulting in vastly different images.

Our coarse-to-fine optimization strategy (last column) produces best results, but it requires using the auto-encoder loss. If this loss is absent, there is no constraint on storing as much information in the coarsest level. As a result, our initialization strategy for multi-level flows (resampling higher levels) does not work well. This alone is however not sufficient as illustrated in the column ($\mathcal{L}_{nll} + \mathcal{L}_{ae}$) where the generative model is trained without the latent-noise loss.

With these experiments it becomes clear that a normalizing flow prior, trained only with the negative log-likelihood loss as suggest in (4) is not sufficient to handle restoration tasks when the images become more complex and of higher resolution.

Generic image restoration. We show that the proposed model is applicable to the restoration of generic images. In order to do so, the model must learn the distribution of patches of high resolution good quality images. For this we use the DIV2K dataset (3) that serves as training and test set for many image quality enhancement works. We use the same train/test split with 800 images in the training set and 100 in the test set. Training is done on random image patches of size 64×64. The normalizing flow architecture used here is very similar to the one described for the Sprites. The number of levels in the architecture is set to $L = 3$ with $K = 4$ steps per level. The main difference is the increased number of intermediate channels in the coupling transforms, from 128 to 256, and the context encoder architecture that is deeper than in the sprites case, with 5 convolutional layer

		DIP	Double-DIP	SRFlow 4x	SRFlow 8x	Ours
JPEG	PSNR↑	28.16	-	27.75	25.27	**30.29**
	SSIM↑	0.85	-	0.80	0.71	**0.86**
	MSSSIM↑	**0.97**	-	0.95	0.91	0.96
	LPIPS↓	**0.17**	-	0.25	0.31	0.23
Noise	PSNR↑	**30.22**	-	27.32	24.73	28.99
	SSIM↑	**0.92**	-	0.81	0.69	0.87
	MSSSIM↑	**0.98**	-	0.96	0.92	0.96
	LPIPS↓	**0.07**	-	0.23	0.34	0.21
Multi Degr.	PSNR↑	26.41	27.62	27.57	25.21	**29.87**
	SSIM↑	0.78	0.80	0.78	0.69	**0.85**
	MSSSIM↑	0.92	0.94	0.94	0.91	**0.96**
	LPIPS↓	0.26	0.27	0.26	0.34	**0.23**

Table 1: Quantitative evaluation on DIV2K.

instead of 1. See the supplementary material for a more detailed description. The normalizing flow model is trained with all the losses indicated in Equation 18.

The restoration of full images of arbitrary size can be done by reconstructing each patch individually. A margin is used to avoid boundary artifacts between patches. More specifically for patches of 64×64 pixels we use a margin $M = 4$ pixels. Neighboring patches overlap in a region of width $2M$ (see supplementary material for illustration). This overlap between adjacent patches yields more consistent results in boundary regions. Restoration results are presented in Figure 5 for different image degradations.

Comparison with Deep Image Prior (DIP) (29). We first compare the two methods on the images presented in the original DIP paper (29). We use our same model trained on the DIV2K dataset. We show competitive restoration results (Figure 1), producing even visually more pleasing reconstruction than DIP on some regions (such as the text and the mouth). The main limit in our case is the patch size used during training. Because of this, it is not possible to inpaint large masked regions. Interestingly however, in this case background regions are better denoised.

We also conduct a quantitative evaluation with results presented in Table 1. Using the test set from DIV2K, we try to restore different degradations: JPEG artifacts, Noise ($\mathcal{N}(0, 5)$) and a combination of artifact removal, denoising and inpainting. For this comparison it is unclear how to best set the number of iterations for the DIP. To handle this, we started from the observation that our method converges to the result in approximately 1 hour of computation. Using the DIP online implementation, this corresponds to around 10k optimization steps on the denoising task. We used this maximum number of steps as the threshold for all images and degradations of the test set. The evaluation demonstrates that our approach is able to achieve competitive results and even outperform DIP on some of the restoration tasks.

Besides achieving competitive results with respect to DIP (29), a key advantage of our approach is that it requires less manual tuning: the DIP requires careful setting of the number of optimization steps to restore a high quality image, and utilizes a specially designed network for each restoration problem. Contrary to our solution that uses the same normalizing flow model across all restoration tasks. Moreover we did not note any convergence issue with between the data term and the prior is set manually, we found that in practice the method is not overly sensitive to changes of this factor (e.g. $\lambda = 50$ was used for all DIV2K examples).

Comparison with Double-DIP As our method significantly outperforms DIP (29) on images with multiple degradations, we additionally compare our method against the more recent Double-DIP (12) by reformulating inpainting as a watermark removal task. Table 1 shows that our method also performs better than Double-DIP (12) in the multi degradation setting.

Comparison with SRFlow We compare our method against SRFlow (19) on the DIV2K dataset. For this comparison we use the SRFlow's 4x and the 8x pretrained models provided by the authors. The restoration is performed as described in (19), Section 4.5, with temperature $\tau = 0.9$. Since SRFlow also operates on fixed-sized patches, we apply the same tiling procedure as for our model. Figure 5 shows that the 8x model produces noticeably more blurry images but does a better job of inpainting the missing regions. Table 1 shows that both models perform significantly worse than our model on all the examples we tested. It is important to note however the runtime difference as their results are obtained with two passes through the network while ours require a costly optimization for each patch.

6. Conclusion

In this paper, we explored using normalizing flows for capturing the distribution of target high quality content to serve as a prior in generic image restoration. This is different from existing learning based priors such as denoising autoencoders or the regularizing properties of convolutional image generator. To the best of our knowledge, this is the first time normalizing flows are successfully used as prior for generic high resolution image restoration tasks. One advantage of this formulation is the learned bijective mapping from image to latent space that we use to express the MAP problem of image reconstruction in latent space. We propose a set of training losses that help structure the base distribution space for our optimization problem, and their importance is demonstrated through the ablation study. Finally, we present experimental results illustrating the capacity of the proposed solution to handle different degradations on data sets of varying complexity. We believe this is an exciting new direction with a lot of potential for future works that would extend to other types of image enhancement problems such as deblurring, super-resolution, dehazing, etc.

References

[1] America In Color. https://www.smithsonianchannel.com/shows/america-in-color/1004516. Accessed: 2018-03-12.

[2] Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3165–3173, 2019.

[3] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1122–1131. IEEE Computer Society, 2017.

[4] Muhammad Asim, Ali Ahmed, and Paul Hand. Invertible generative models for inverse problems: mitigating representation error and dataset bias. In *ICML*, pages 0–0, 2020.

[5] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems*, pages 284–293, 2019.

[6] Siavash Arjomand Bigdeli, Matthias Zwicker, Paolo Favaro, and Meiguang Jin. Deep mean-shift priors for image restoration. In *Advances in Neural Information Processing Systems*, pages 763–772, 2017.

[7] Zezhou Cheng, Matheus Gadelha, Subhransu Maji, and Daniel Sheldon. A bayesian perspective on the deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5443–5451, 2019.

[8] Victor Cornillère, Abdelaziz Djelouah, Wang Yifan, Olga Sorkine-Hornung, and Christopher Schroers. Blind image super resolution with spatially variant degradations. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)*, 2019.

[9] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

[10] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[11] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006 Papers*, pages 787–794. 2006.

[12] Yossi Gandelsman, Assaf Shocher, and Michal Irani. "double-dip": Unsupervised image decomposition via coupled deep-image-priors. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11026–11035. Computer Vision Foundation / IEEE, 2019.

[13] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[16] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 10236–10245, 2018.

[17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[18] Ce Liu, Michael Rubinstein, Mike Krainin, and Bill Freeman. PhotoScan: Taking Glare-Free Pictures of Pictures. https://ai.googleblog.com/2017/04/photoscan-taking-glare-free-pictures-of.html. Accessed: 2020-05-25.

[19] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pages 715–732. Springer, 2020.

[20] Gary Mataev, Peyman Milanfar, and Michael Elad. Deepred: Deep image prior powered by red. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[21] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1781–1790, 2017.

[22] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[23] Haesol Park and Kyoung Mu Lee. Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4613–4621, 2017.

[24] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[25] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June*

13-19, 2020, pages 3338–3347. IEEE, 2020.

[26] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1530–1538, 2015.

[27] JH Rick Chang, Chun-Liang Li, Barnabas Poczos, BVK Vijaya Kumar, and Aswin C Sankaranarayanan. One network to solve them all–solving linear inverse problems using deep projection models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5888–5897, 2017.

[28] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[29] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[30] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 864–873, 2018.

[31] 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.

[32] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

[33] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.