SRFlow: Learning the Super-Resolution Space with Normalizing Flow

Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte

Computer Vision Laboratory, ETH Zurich
{andreas.lugmayr,martin.danelljan,vangool,radu.timofte}@vision.ee.ethz.ch

Abstract. Super-resolution is an ill-posed problem, since it allows for multiple predictions for a given low-resolution image. This fundamental fact is largely ignored by state-of-the-art deep learning based approaches. These methods instead train a deterministic mapping using combinations of reconstruction and adversarial losses. In this work, we therefore propose **SRFlow**: a normalizing flow based super-resolution method capable of learning the conditional distribution of the output given the low-resolution input. Our model is trained in a principled manner using a single loss, namely the negative log-likelihood. SRFlow therefore directly accounts for the ill-posed nature of the problem, and learns to predict diverse photo-realistic high-resolution images. Moreover, we utilize the strong image posterior learned by SRFlow to design flexible image manipulation techniques, capable of enhancing super-resolved images by, e.g., transferring content from other images. We perform extensive experiments on faces, as well as on super-resolution in general. SRFlow outperforms state-of-the-art GAN-based approaches in terms of both PSNR and perceptual quality metrics, while allowing for diversity through the exploration of the space of super-resolved solutions. Code: git.io/Jfpyu

1 Introduction

Single image super-resolution (SR) is an active research topic with several important applications. It aims to enhance the resolution of a given image by adding

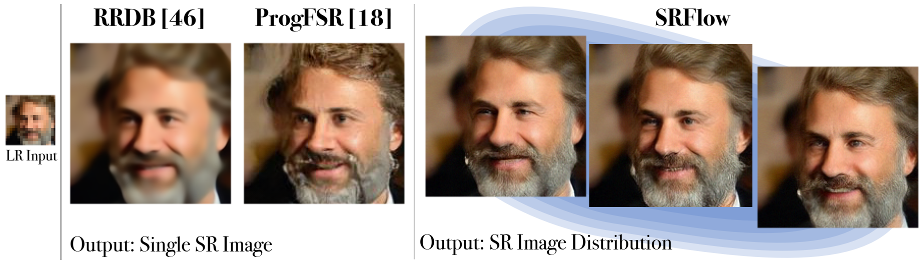


Fig. 1. While prior work trains a deterministic mapping, SRFlow learns the distribution of photo-realistic HR images for a given LR image. This allows us to explicitly account for the ill-posed nature of the SR problem, and to sample diverse images. (8× upscaling)

missing high-frequency information. Super-resolution is therefore a fundamentally ill-posed problem. In fact, for a given low-resolution (LR) image, there exist infinitely many compatible high-resolution (HR) predictions. This poses severe challenges when designing deep learning based super-resolution approaches.

Initial deep learning approaches [11,12,19,21,23] employ feed-forward architectures trained using standard L_2 or L_1 reconstruction losses. While these methods achieve impressive PSNR, they tend to generate blurry predictions. This shortcoming stems from discarding the ill-posed nature of the SR problem. The employed L_2 and L_1 reconstruction losses favor the prediction of an *average* over the plausible HR solutions, leading to the significant reduction of high-frequency details. To address this problem, more recent approaches [2,15,22,38,46,53] integrate adversarial training and perceptual loss functions. While achieving sharper images with better perceptual quality, such methods only predict a *single* SR output, which does not fully account for the ill-posed nature of the SR problem.

We address the limitations of the aforementioned approaches by learning the conditional *distribution* of plausible HR images given the input LR image. To this end, we design a conditional normalizing flow [10,37] architecture for image super-resolution. Thanks to the exact log-likelihood training enabled by the flow formulation, our approach can model expressive distributions over the HR image space. This allows our network to learn the generation of photo-realistic SR images that are consistent with the input LR image, without any additional constraints or losses. Given an LR image, our approach can sample multiple diverse SR images from the learned distribution. In contrast to conventional methods, our network can thus explore the space of SR images (see Fig. 1).

Compared to standard Generative Adversarial Network (GAN) based SR approaches [22,46], the proposed flow-based solution exhibits a few key advantages. First, our method naturally learns to generate diverse SR samples without suffering from mode-collapse, which is particularly problematic in the conditional GAN setting [17,29]. Second, while GAN-based SR networks require multiple losses with careful parameter tuning, our network is stably trained with a single loss: the negative log-likelihood. Third, the flow network employs a fully invertible encoder, capable of mapping any input HR image to the latent flow-space and ensuring *exact* reconstruction. This allows us to develop powerful image manipulation techniques for editing the predicted SR or any existing HR image.

Contributions: We propose **SRFlow**, a flow-based super-resolution network capable of accurately learning the distribution of realistic HR images corresponding to the input LR image. In particular, the main contributions of this work are as follows: **(i)** We are the first to design a conditional normalizing flow architecture that achieves state-of-the-art super-resolution quality. **(ii)** We harness the strong HR distribution learned by SRFlow to develop novel techniques for controlled image manipulation and editing. **(iii)** Although only trained for super-resolution, we show that SRFlow is capable of image denoising and restoration. **(iv)** Comprehensive experiments for face and general image super-resolution show that our approach outperforms state-of-the-art GAN-based methods for both perceptual and reconstruction-based metrics.

2 Related Work

Single image SR: Super-resolution has long been a fundamental challenge in computer vision due to its ill-posed nature. Early learning-based methods mainly employed sparse coding based techniques [8,41,51,52] or local linear regression [43,45,49]. The effectiveness of example-based deep learning for super-resolution was first demonstrated by SRCNN [11], which further led to the development of more effective network architectures [12,19,21,23]. However, these methods do not reproduce the sharp details present in natural images due to their reliance on L_2 and L_1 reconstruction losses. This was addressed in URDGN [53], SRGAN [22] and more recent approaches [2,15,38,46] by adopting a conditional GAN based architecture and training strategy. While these works aim to predict *one* example, we undertake the more ambitious goal of learning the distribution of *all* plausible reconstructions from the natural image manifold.

Stochastic SR: The problem of generating diverse super-resolutions has received relatively little attention. This is partly due to the challenging nature of the problem. While GANs provide an method for learning a distribution over data [14], conditional GANs are known to be extremely susceptible to mode collapse since they easily learn to ignore the stochastic input signal [17,29]. Therefore, most conditional GAN based approaches for super-resolution and image-to-image translation resort to purely deterministic mappings [22,35,46]. A few recent works [4,7,30] address GAN-based stochastic SR by exploring techniques to avoid mode collapse and explicitly enforcing low-resolution consistency. In contrast to those works, we design a flow-based architecture trained using the negative log-likelihood loss. This allows us to learn the conditional distribution of HR images, without any additional constraints, losses, or post-processing techniques to enforce low-resolution consistency. A different line of research [6,39,40] exploit the internal patch recurrence by only training the network on the input image itself. Recently [39] employed this strategy to learn a GAN capable of stochastic SR generation. While this is an interesting direction, our goal is to exploit large image datasets to learn a general distribution over the image space.

Normalizing flow: Generative modelling of natural images poses major challenges due to the high dimensionality and complex structure of the underlying data distribution. While GANs [14] have been explored for several vision tasks, Normalizing Flow based models [9,10,20,37] have received much less attention. These approaches parametrize a complex distribution $p_{\mathbf{y}}(\mathbf{y}|\boldsymbol{\theta})$ using an invertible neural network $f_{\boldsymbol{\theta}}$, which maps samples drawn from a simple (e.g. Gaussian) distribution $p_{\mathbf{z}}(\mathbf{z})$ as $\mathbf{y} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z})$. This allows the *exact* negative log-likelihood $-\log p_{\mathbf{y}}(\mathbf{y}|\boldsymbol{\theta})$ to be computed by applying the change-of-variable formula. The network can thus be trained by directly minimizing the negative log-likelihood using standard SGD-based techniques. Recent works have investigated conditional flow models for point cloud generation [36,50] as well as class [24] and image [3,48] conditional generation of images. The latter works [3,48] adapt the widely successful Glow architecture [20] to conditional image generation by concatenating the encoded conditioning variable in the affine coupling layers [9,10].

The concurrent work [48] consider the SR task as an example application, but only addressing 2× magnification and without comparisons with state-of-the-art GAN-based methods. While we also employ the conditional flow paradigm for its theoretically appealing properties, our work differs from these previous approaches in several aspects. Our work is first to develop a conditional flow architecture for SR that provides favorable or superior results compared to state-of-the-art GAN-based methods. Second, we develop powerful flow-based image manipulation techniques, applicable for guided SR and to editing existing HR images. Third, we introduce new training and architectural considerations. Lastly, we demonstrate the generality and strength of our learned image posterior by applying SRFlow to image restoration tasks, unseen during training.

3 Proposed Method: SRFlow

We formulate super-resolution as the problem of learning a conditional probability distribution over high-resolution images, given an input low-resolution image. This approach explicitly addresses the ill-posed nature of the SR problem by aiming to capture the full diversity of possible SR images from the natural image manifold. To this end, we design a conditional normalizing flow architecture, allowing us to learn rich distributions using exact log-likelihood based training.

3.1 Conditional Normalizing Flows for Super-Resolution

The goal of super-resolution is to predict higher-resolution versions $\mathbf{y}$ of a given low-resolution image $\mathbf{x}$ by generating the absent high-frequency details. While most current approaches learn a deterministic mapping $\mathbf{x} \mapsto \mathbf{y}$, we aim to capture the full conditional distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ of natural HR images $\mathbf{y}$ corresponding to the LR image $\mathbf{x}$. This constitutes a more challenging task, since the model must span a variety of possible HR images, instead of just predicting a single SR output. Our intention is to train the parameters $\boldsymbol{\theta}$ of the distribution in a purely data-driven manner, given a large set of LR-HR training pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{M}$.

The core idea of normalizing flow [9,37] is to parametrize the distribution $p_{\mathbf{y}|\mathbf{x}}$ using an invertible neural network $f_{\boldsymbol{\theta}}$. In the conditional setting, $f_{\boldsymbol{\theta}}$ maps an HR-LR image pair to a latent variable $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{y}; \mathbf{x})$. We require this function to be invertible w.r.t. the first argument $\mathbf{y}$ for any LR image $\mathbf{x}$. That is, the HR image $\mathbf{y}$ can always be exactly reconstructed from the latent encoding $\mathbf{z}$ as $\mathbf{y} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z}; \mathbf{x})$. By postulating a simple distribution $p_{\mathbf{z}}(\mathbf{z})$ (e.g. a Gaussian) in the latent space $\mathbf{z}$, the conditional distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ is implicitly defined by the mapping $\mathbf{y} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z}; \mathbf{x})$ of samples $\mathbf{z} \sim p_{\mathbf{z}}$. The key aspect of normalizing flows is that the probability density $p_{\mathbf{y}|\mathbf{x}}$ can be explicitly computed as,

$$p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = p_{\mathbf{z}}\big(f_{\boldsymbol{\theta}}(\mathbf{y}; \mathbf{x})\big) \left| \det \frac{\partial f_{\boldsymbol{\theta}}}{\partial \mathbf{y}}(\mathbf{y}; \mathbf{x}) \right| . \tag{1}$$

It is derived by applying the change-of-variables formula for densities, where the second factor is the resulting volume scaling given by the determinant of the

Jacobian $\frac{\partial f_{\boldsymbol{\theta}}}{\partial \mathbf{y}}$. The expression (1) allows us to train the network by minimizing the negative log-likelihood (NLL) for training samples pairs $(\mathbf{x}, \mathbf{y})$,

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y}) = -\log p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = -\log p_{\mathbf{z}}\big(f_{\boldsymbol{\theta}}(\mathbf{y}; \mathbf{x})\big) - \log \left| \det \frac{\partial f_{\boldsymbol{\theta}}}{\partial \mathbf{y}}(\mathbf{y}; \mathbf{x}) \right| . \quad (2)$$

HR image samples $\mathbf{y}$ from the learned distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ are generated by applying the inverse network $\mathbf{y} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z}; \mathbf{x})$ to random latent variables $\mathbf{z} \sim p_{\mathbf{z}}$.

In order to achieve a tractable expression of the second term in (2), the neural network $f_{\boldsymbol{\theta}}$ is decomposed into a sequence of N invertible layers $\mathbf{h}^{n+1} = f_{\boldsymbol{\theta}}^n(\mathbf{h}^n; g_{\boldsymbol{\theta}}(\mathbf{x}))$, where $\mathbf{h}^0 = \mathbf{y}$ and $\mathbf{h}^N = \mathbf{z}$. We let the LR image to first be encoded by a shared deep CNN $g_{\boldsymbol{\theta}}(\mathbf{x})$ that extracts a rich representation suitable for conditioning in all flow-layers, as detailed in Sec. 3.3. By applying the chain rule along with the multiplicative property of the determinant [10], the NLL objective in (2) can be expressed as

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y}) = -\log p_{\mathbf{z}}(\mathbf{z}) - \sum_{n=0}^{N-1} \log \left| \det \frac{\partial f_{\boldsymbol{\theta}}^n}{\partial \mathbf{h}^n}(\mathbf{h}^n; g_{\boldsymbol{\theta}}(\mathbf{x})) \right| . \quad (3)$$

We thus only need to compute the log-determinant of the Jacobian $\frac{\partial f_{\boldsymbol{\theta}}^n}{\partial \mathbf{h}^n}$ for each individual flow-layer $f_{\boldsymbol{\theta}}^n$. To ensure efficient training and inference, the flow layers $f_{\boldsymbol{\theta}}^n$ thus need to allow efficient inversion and a tractable Jacobian determinant. This is further discussed next, where we detail the employed conditional flow layers $f_{\boldsymbol{\theta}}^n$ in our SR architecture. Our overall network architecture for flow-based super-resolution is depicted in Fig. 2.

3.2 Conditional Flow Layers

The design of flow-layers $f_{\boldsymbol{\theta}}^n$ requires care in order to ensure a well-conditioned inverse and a tractable Jacobian determinant. This challenge was first addressed in [9,10] and has recently spurred significant interest [5,13,20]. We start from the unconditional Glow architecture [20], which is itself based on the RealNVP [10]. The flow layers employed in these architectures can be made conditional in a straight-forward manner [3,48]. We briefly review them here along with our introduced Affine Injector layer.

Conditional Affine Coupling: The affine coupling layer [9,10] provides a simple and powerful strategy for constructing flow-layers that are easily invertible. It is trivially extended to the conditional setting as follows,

$$\mathbf{h}_A^{n+1} = \mathbf{h}_A^n \,, \qquad \mathbf{h}_B^{n+1} = \exp\big(f_{\boldsymbol{\theta},\mathrm{s}}^n(\mathbf{h}_A^n; \mathbf{u})\big) \cdot \mathbf{h}_B^n + f_{\boldsymbol{\theta},\mathrm{b}}^n(\mathbf{h}_A^n; \mathbf{u}) \,. \quad (4)$$

Here, $\mathbf{h}^n = (\mathbf{h}_A^n, \mathbf{h}_B^n)$ is a partition of the activation map in the channel dimension. Moreover, $\mathbf{u}$ is the conditioning variable, set to the encoded LR image $\mathbf{u} = g_{\boldsymbol{\theta}}(\mathbf{x})$ in our work. Note that $f_{\boldsymbol{\theta},\mathrm{s}}^n$ and $f_{\boldsymbol{\theta},\mathrm{b}}^n$ represent *arbitrary* neural networks that generate the scaling and bias of $\mathbf{h}_B^n$. The Jacobian of (4) is triangular, enabling the efficient computation of its log-determinant as $\sum_{ijk} f_{\boldsymbol{\theta},\mathrm{s}}^n(\mathbf{h}_A^n; \mathbf{u})_{ijk}$.

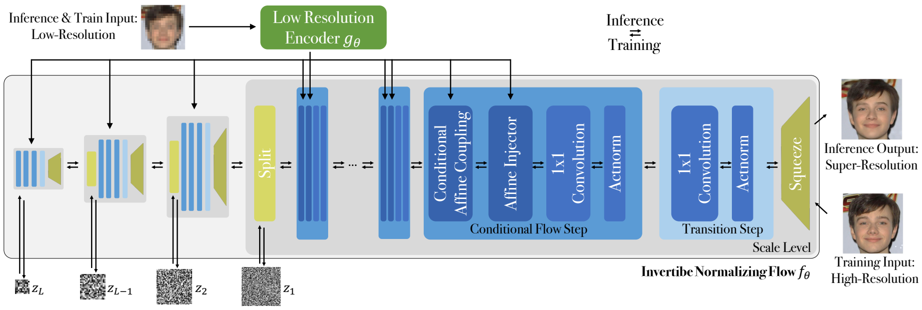


Fig. 2. SRFlow's conditional normalizing flow architecture. Our model consists of an invertible flow network $f_{\boldsymbol{\theta}}$, conditioned on an encoding (green) of the low-resolution image. The flow network operates at multiple scale levels (gray). The input is processed through a series of flow-steps (blue), each consisting of four different layers. Through exact log-likelihood training, our network learns to transform a Gaussian density $p_{\mathbf{z}}(\mathbf{z})$ to the conditional HR-image distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$. During training, an LR-HR $(\mathbf{x},\mathbf{y})$ image pair is input in order to compute the negative log-likelihood loss. During inference, the network operates in the reverse direction by inputting the LR image along with a random variables $\mathbf{z} = (\mathbf{z}_l)_{l=1}^{L} \sim p_{\mathbf{z}}$, which generates sample SR images from the learned distribution $p_{\mathbf{y}|\mathbf{x}}$.

Invertible 1×1 **Convolution:** General convolutional layers are often intractable to invert or evaluate the determinant of. However, [20] demonstrated that a 1×1 convolution $\mathbf{h}_{ij}^{n+1} = W\mathbf{h}_{ij}^{n}$ can be efficiently integrated since it acts on each spatial coordinate (i,j) independently, which leads to a block-diagonal structure. We use the non-factorized formulation in [20].

Actnorm: This provides a channel-wise normalization through a learned scaling and bias. We keep this layer in its standard un-conditional form [20].

Squeeze: It is important to process the activations at different scales in order to capture correlations and structures over larger distances. The squeeze layer [20] provides an invertible means to halving the resolution of the activation map $\mathbf{h}^n$ by reshaping each spatial 2×2 neighborhood into the channel dimension.

Affine Injector: To achieve more direct information transfer from the low-resolution image encoding $\mathbf{u} = g_{\boldsymbol{\theta}}(\mathbf{x})$ to the flow branch, we additionally introduce the affine injector layer. In contrast to the conditional affine coupling layer, our affine injector layer directly affects all channels and spatial locations in the activation map $\mathbf{h}^n$. This is achieved by predicting an element-wise scaling and bias using only the conditional encoding $\mathbf{u}$,

$$\mathbf{h}^{n+1} = \exp\big(f_{\boldsymbol{\theta},\mathrm{s}}^{n}(\mathbf{u})\big) \cdot \mathbf{h}^{n} + f_{\boldsymbol{\theta},\mathrm{b}}(\mathbf{u})\,. \tag{5}$$

Here, $f_{\boldsymbol{\theta},\mathrm{s}}$ and $f_{\boldsymbol{\theta},\mathrm{s}}$ can be any network. The inverse of (5) is trivially obtained as $\mathbf{h}^n = \exp(-f_{\boldsymbol{\theta},\mathrm{s}}^{n}(\mathbf{u})) \cdot (\mathbf{h}^{n+1} - f_{\boldsymbol{\theta},\mathrm{b}}^{n}(\mathbf{u}))$ and the log-determinant is given by $\sum_{ijk} f_{\boldsymbol{\theta},\mathrm{s}}^{n}(\mathbf{u})_{ijk}$. Here, the sum ranges over all spatial i,j and channel indices k.

3.3 Architecture

Our SRFlow architecture, depicted in Fig. 2, consists of the invertible flow network $f_{\boldsymbol{\theta}}$ and the LR encoder $g_{\boldsymbol{\theta}}$. The flow network is organized into L levels, each operating at a resolution of $\frac{H}{2^l} \times \frac{W}{2^l}$, where $l \in \{1, \dots, L\}$ is the level number and $H \times W$ is the HR resolution. Each level itself contains K number of flow-steps.

Flow-step: Each flow-step in our approach consists of four different layers, as visualized in Fig. 2. The Actnorm if applied first, followed by the 1×1 convolution. We then apply the two conditional layers, first the Affine Injector followed by the Conditional Affine Coupling.

Level transitions: Each level first performs a squeeze operation that effectively halves the spatial resolution. We observed that this layer can lead to checkerboard artifacts in the reconstructed image, since it is only based on pixel re-ordering. To learn a better transition between the levels, we therefore remove the conditional layers first few flow steps after the squeeze (see Fig. 2). This allows the network to learn a linear invertible interpolation between neighboring pixels. Similar to [20], we split off 50% of the channels before the next squeeze layer. Our latent variables $(z_l)_{l=1}^{L}$ thus model variations in the image at different resolutions, as visualized in Fig. 2.

Low-resolution encoding network $g_{\boldsymbol{\theta}}$: SRFlow allows for the use of any differentiable architecture for the LR encoding network $g_{\boldsymbol{\theta}}$, since it does not need to be invertible. Our approach can therefore benefit from the advances in standard feed-forward SR architectures. In particular, we adopt the popular CNN architecture based on Residual-in-Residual Dense Blocks (RRDB) [46], which builds upon [22,23]. It employs multiple residual and dense skip connections, without any batch normalization layers. We first discard the final upsampling layers in the RRDB architecture since we are only interested in the underlying representation and not the SR prediction. In order to capture a richer representation of the LR image at multiple levels, we additionally concatenate the activations after each RRDB block to form the final output of $g_{\boldsymbol{\theta}}$.

Details: We employ $K = 16$ flow-steps at each level, with two additional unconditional flow-steps after each squeeze layer (discussed above). We use $L = 3$ and $L = 4$ levels for SR factors 4× and 8× respectively. For general image SR, we use the standard 23-block RRDB architecture [46] for the LR encoder $g_{\boldsymbol{\theta}}$. For faces, we reduce to 8 blocks for efficiency. The networks $f^n_{\boldsymbol{\theta},\mathrm{s}}$ and $f^n_{\boldsymbol{\theta},\mathrm{b}}$ in the conditional affine coupling (4) and the affine injector (5) are constructed using two shared convolutional layers with ReLU, followed by a final convolution.

3.4 Training Details

We train our entire SRFlow network using the negative log-likelihood loss (3). We sample batches of 16 LR-HR image pairs $(\mathbf{x}, \mathbf{y})$. During training, we use an HR patch size of 160×160. As optimizer we use Adam with a starting learning rate of $5 \cdot 10^{-4}$, which is halved at $50\%, 75\%, 90\%$ and 95% of the total training iterations. To increase training efficiency, we first pre-train the LR encoder $g_{\boldsymbol{\theta}}$ using an L_1 loss for 200k iterations. We then train our full SRFlow architecture

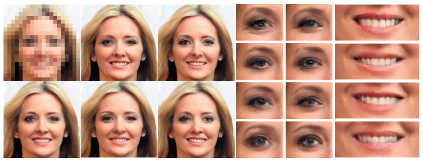


Fig. 3. Random 8× SR samples generated by SRFlow using a temperature $\tau = 0.8$. LR image is shown in top left.

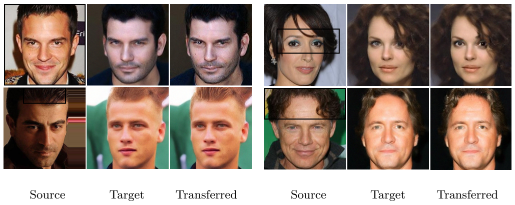


Fig. 4. Latent space transfer from the region marked by the box to the target image. (8×)

using only the loss (3) for 200k iterations. Our network takes 5 days to train on a single NVIDIA V100 GPU. Further details are provided in the supplementary.

Datasets: For face super-resolution, we use the CelebA [25] dataset. Similar to [20,18], we pre-process the dataset by cropping aligned patches, which are resized to the HR resolution of 160 × 160. We employ the full train split (160k images). For general SR, we use the same training data as ESRGAN [46], consisting of the train split of 800 DIV2k [1] along with 2650 images from Flickr2K. The LR images are constructed using the standard MATLAB bicubic kernel.

4 Applications and Image Manipulations

In this section, we explore the use of our SRFlow network for a variety of applications and image manipulation tasks. Our techniques exploit two key advantages of our SRFlow network, which are not present in GAN-based super-resolution approaches [46]. First, our network models a distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ in HR-image space, instead of only predicting a single image. It therefore possesses great flexibility by capturing a variety of possible HR predictions. This allows different predictions to be explored using additional guiding information or random sampling. Second, the flow network $f_{\boldsymbol{\theta}}(\mathbf{y}; \mathbf{x})$ is a fully invertible encoder-decoder. Hence, *any* HR image $\tilde{\mathbf{y}}$ can be encoded into the latent space as $\tilde{\mathbf{z}} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}; \mathbf{x})$ and *exactly* reconstructed as $\tilde{\mathbf{y}} = f_{\boldsymbol{\theta}}^{-1}(\tilde{\mathbf{z}}; \mathbf{x})$. This bijective correspondence allows us to flexibly operate in both the latent and image space.

4.1 Stochastic Super-resolution

The distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ learned by our SRFlow can be explored by sampling different SR predictions as $\mathbf{y}^{(i)} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z}^{(i)}; \mathbf{x})$, $\mathbf{z}^{(i)} \sim p_{\mathbf{z}}$ for a given LR image $\mathbf{x}$. As commonly observed for flow-based models, the best results are achieved when sampling with a slightly lower variance [20]. We therefore use a Gaussian $\mathbf{z}^{(i)} \sim \mathcal{N}(0, \tau)$ with variance τ (also called temperature). Results are visualized in Fig. 3 for $\tau = 0.8$. Our approach generates a large variety of SR images, including differences in e.g. hair and facial attributes, while preserving consistency with the LR image. Since our latent variables $\mathbf{z}_{ijkl}$ are spatially localized, specific parts can be re-sampled, enabling more controlled interactive editing and exploration of the SR image.

4.2 LR-Consistent Style Transfer

Our SRFlow allows transferring the style of an existing HR image $\tilde{\mathbf{y}}$ when super-resolving an LR image $\mathbf{x}$. This is performed by first encoding the source HR image as $\tilde{\mathbf{z}} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}; d_{\downarrow}(\tilde{\mathbf{y}}))$, where $d_{\downarrow}$ is the down-scaling operator. The encoding $\tilde{\mathbf{z}}$ can then be used to as the latent variable for the super-resolution of $\mathbf{x}$ as $\mathbf{y} = f_{\boldsymbol{\theta}}^{-1}(\tilde{\mathbf{z}}; \mathbf{x})$. This operation can also be performed on local regions of the image. Examples in Fig. 4 show the transfer in the style of facial characteristics, hair and eye color. Our SRFlow network automatically aims to ensure consistency with the LR image without any additional constraints.

4.3 Latent Space Normalization

We develop more advanced image manipulation techniques by taking advantage of the invertability of the SRFlow network $f_{\boldsymbol{\theta}}$ and the learned super-resolution posterior. The core idea of our approach is to map any HR image containing desired content to the latent space, where the latent statistics can be normalized in order to make it consistent with the low-frequency information in the given LR image. Let $\mathbf{x}$ be a low-resolution image and $\tilde{\mathbf{y}}$ be *any* high-resolution image, not necessarily consistent with the LR image $\mathbf{x}$. For example, $\tilde{\mathbf{y}}$ can be an edited version of a super-resolved image or a guiding image for the super-resolution image. Our goal is to achieve an HR image $\mathbf{y}$, containing image content from $\tilde{\mathbf{y}}$, but that is consistent with the LR image $\mathbf{x}$.

The latent encoding for the given image pair is computed as $\tilde{\mathbf{z}} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}; \mathbf{x})$. Note that our network is trained to predict consistent and natural SR images for latent variables sampled from a standard Gaussian distribution $p_{\mathbf{z}} = \mathcal{N}(0, I)$. Since $\tilde{\mathbf{y}}$ is not necessarily consistent with the LR image $\mathbf{x}$, the latent variables $\tilde{\mathbf{z}}_{ijkl}$ do not have the same statistics as if independently sampled from $\mathbf{z}_{ijkl} \sim \mathcal{N}(0, \tau)$. Here, τ denotes an additional temperature scaling of the desired latent distribution. In order to achieve desired statistics, we normalize the first two moments of collections of latent variables. In particular, if $\{z_i\}_1^N \sim \mathcal{N}(0, \tau)$ are independent, then it is well known [33] that their empirical mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ are distributed according to,

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} z_i \sim \mathcal{N}\left(0, \frac{\tau}{N}\right), \;\; \hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (z_i - \hat{\mu})^2 \sim \mathit{\Gamma}\left(\frac{N-1}{2}, \frac{2\tau}{N-1}\right). \quad (6)$$

Here, $\mathit{\Gamma}(k, \theta)$ is a gamma distribution with shape and scale parameters k and θ respectively. For a given collection $\tilde{\mathcal{Z}} \subset \{\mathbf{z}_{ijkl}\}$ of latent variables, we normalize their statistics by first sampling a new mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ according to (6), where $N = |\tilde{\mathcal{Z}}|$ is the size of the collection. The latent variables in the collection are then normalized as,

$$\hat{z} = \frac{\hat{\sigma}}{\tilde{\sigma}}(\tilde{z} - \tilde{\mu}) + \hat{\mu}, \quad \forall \tilde{z} \in \tilde{\mathcal{Z}}. \quad (7)$$

Here, $\tilde{\mu}$ and $\tilde{\sigma}^2$ denote the empirical mean and variance of the collection $\tilde{\mathcal{Z}}$.

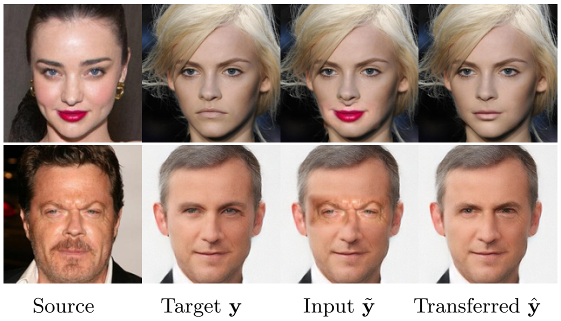


Fig. 5. Image content transfer for an existing HR image (top) and an SR prediction (bottom). Content from the source is applied directly to the target. By applying latent space normalization in our SRFlow, the content is integrated and harmonized.

		Original	Super-Resloved	Restored
DIV2K	PSNR↑	22.48	23.19	27.81
	SSIM↑	0.49	0.51	0.73
	LPIPS↓	0.370	0.364	0.255
CelebA	PSNR↑	22.52	24.25	27.62
	SSIM↑	0.48	0.63	0.78
	LPIPS↓	0.326	0.172	0.143

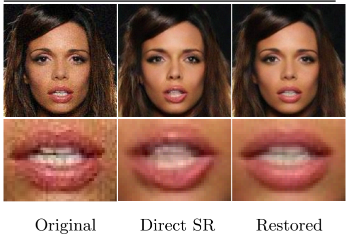


Fig. 6. Comparision of super-resolving the LR of the original and normalizing the latent space for image restoration.

The normalization in (7) can be performed using different collections $\tilde{\mathcal{Z}}$. We consider three different strategies in this work. **Global normalization** is performed over the entire latent space, using $\tilde{\mathcal{Z}} = \{\mathbf{z}_{ijkl}\}_{ijkl}$. For **local normalization**, each spatial position i, j in each level l is normalized independently as $\tilde{\mathcal{Z}}_{ijl} = \{\mathbf{z}_{ijkl}\}_k$. This better addresses cases where the statistics is spatially varying. **Spatial normalization** is performed independently for each feature channel k and level l, using $\tilde{\mathcal{Z}}_{kl} = \{\mathbf{z}_{ijkl}\}_{ij}$. It addresses global effects in the image that activates certain channels, such as color shift or noise. In all three cases, normalized latent variable $\hat{\mathbf{z}}$ is obtained by applying (7) for all collections, which is an easily parallelized computation. The final HR image is then reconstructed as $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}^{-1}(\hat{\mathbf{z}}, \mathbf{x})$. Note that our normalization procedure is stochastic, since a new mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ are sampled independently for every collection of latent variables $\tilde{\mathcal{Z}}$. This allows us to sample from the natural diversity of predictions $\hat{\mathbf{y}}$, that integrate content from $\tilde{\mathbf{y}}$. Next, we explore our latent space normalization technique for different applications.

4.4 Image Content Transfer

Here, we aim to manipulate an HR image by transferring content from other images. Let $\mathbf{x}$ be an LR image and $\mathbf{y}$ a corresponding HR image. If we are manipulating a super-resolved image, then $\mathbf{y} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z}, \mathbf{x})$ is an SR sample of $\mathbf{x}$. However, we can also manipulate an existing HR image $\mathbf{y}$ by setting $\mathbf{x} = d_{\downarrow}(\mathbf{y})$ to the down-scaled version of $\mathbf{y}$. We then manipulate $\mathbf{y}$ directly in the image space by simply inserting content from other images, as visualized in Fig. 5. To harmonize the resulting manipulated image $\tilde{\mathbf{y}}$ by ensuring consistency with the LR image $\mathbf{x}$, we compute the latent encoding $\tilde{\mathbf{z}} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}; \mathbf{x})$ and perform *local normalization* of the latent variables as described in Sec. 4.3. We only normalize the affected regions of the image in order to preserve the non-manipulated content. Results are shown in Fig. 5. If desired, the emphasis on LR-consistency can be reduced by training SRFlow with randomly misaligned HR-LR pairs, which allows increased manipulation flexibility (see supplement).

4.5 Image Restoration

We demonstrate the strength of our learned image posterior by applying it for image restoration tasks. Note that we here employ the *same* SRFlow network, that is trained only for super-resolution, and not for the explored tasks. In particular, we investigate degradations that mainly affect the high frequencies in the image, such as noise and compression artifacts. Let $\tilde{\mathbf{y}}$ be a degraded image. Noise and other high-frequency degradations are largely removed when down-sampled $\mathbf{x} = d_{\downarrow}(\tilde{\mathbf{y}})$. Thus a cleaner image can be obtained by applying any super-resolution method to $\mathbf{x}$. However, this generates poor results since important image information is lost in the down-sampling process (Fig. 6, center).

Our approach can go beyond this result by directly utilizing the original image $\tilde{\mathbf{y}}$. The degraded image along with its down-sampled variant are input to our SRFlow network to generate the latent variable $\tilde{\mathbf{z}} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}; \mathbf{x})$. We then perform first *spatial* and then *local* normalization of $\tilde{\mathbf{z}}$, as described in Sec. 4.3. The restored image is then predicted as $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}^{-1}(\hat{\mathbf{z}}, \mathbf{x})$. By, denoting the normalization operation as $\hat{\mathbf{z}} = \phi(\tilde{\mathbf{z}})$, the full restoration mapping can be expressed as $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}^{-1}(\phi(f_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}; d_{\downarrow}(\tilde{\mathbf{y}}))), d_{\downarrow}(\tilde{\mathbf{y}}))$. As shown visually and quantitatively in Fig. 6, this allows us to recover a substantial amount of details from the original image Intuitively, our approach works by mapping the degraded image $\tilde{\mathbf{y}}$ to the *closest* image within the learned distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. Since SRFlow is not trained with such degradations, $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ mainly models *clean* images. Our normalization therefore automatically restores the image when it is transformed to a more *likely* image according to our SR distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$.

5 Experiments

We perform comprehensive experiments for super-resolution of faces and of generic images in comparisons with current state-of-the-art and an ablative analysis. Applications, such as image manipulation tasks, are presented in Sec. 4, with additional results, analysis and visuals in the supplement.

Evaluation Metrics: To evaluate the perceptual distance to the Ground Truth, we report the default LPIPS [54]. It is a learned distance metric, based on the feature-space of a finetuned AlexNet. We report the standard fidelity oriented metrics, Peak Signal to Noise Ratio (PSNR) and structural similarity index (SSIM) [47], although they are known to not correlate well with the human perception of image quality [16,22,26,28,42,44]. Furthermore, we report the no-reference metrics NIQE [32], BRISQUE [31] and PIQUE [34]. In addition to visual quality, consistency with the LR image is an important factor. We therefore evaluate this aspect by reporting the LR-PSNR, computed as the PSNR between the downsampled SR image and the original LR image.

Table 1. Results for 8× SR of faces of CelebA. We compare using both the standard bicubic kernel and the progressive linear kernel from [18]. We also report the diversity in the SR output in terms of the pixel standard deviation σ.

LR		↑PSNR	↑SSIM	↓LPIPS	↑LR-PSNR	↓NIQE	↓BRISQUE	↓PIQUE	↑Diversity σ
Bicubic	Bicubic	23.15	0.63	0.517	35.19	7.82	58.6	99.97	0
	RRDB [46]	26.59	0.77	0.230	48.22	6.02	49.7	86.5	0
	ESRGAN [46]	22.88	0.63	0.120	34.04	3.46	23.7	32.4	0
	SRFlow $\tau = 0.8$	25.24	0.71	0.110	50.85	4.20	23.2	24.0	5.21
Prog.	ProgFSR [18]	23.97	0.67	0.129	41.95	3.49	28.6	33.2	0
	SRFlow $\tau = 0.8$	25.20	0.71	0.110	51.05	4.20	22.5	23.1	5.28

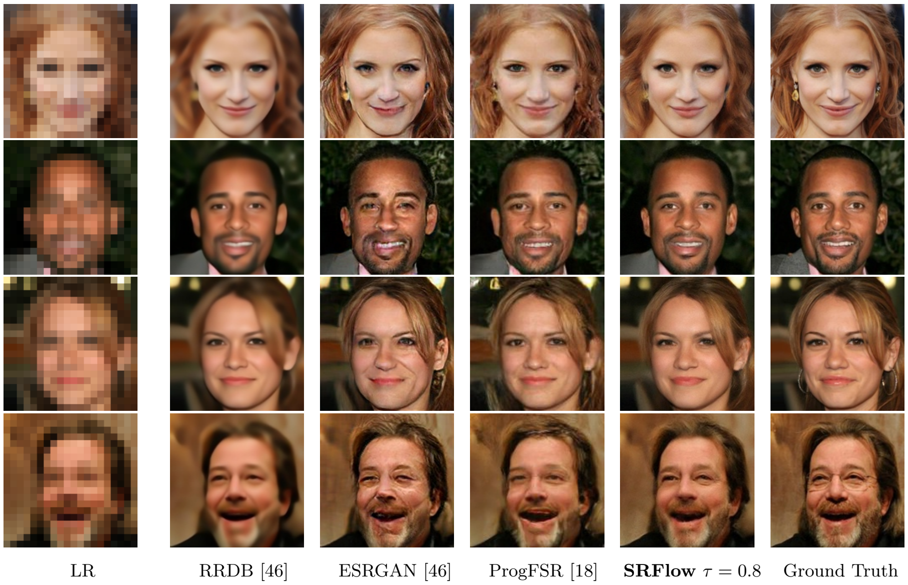


Fig. 7. Comparison of our SRFlow with state-of-the-art for 8× face SR on CelebA.

5.1 Face Super-Resolution

We evaluate SRFlow for face SR (8×) using 5000 images from the test split of the CelebA dataset. We compare with bicubic, RRDB [46], ESRGAN [46], and ProgFSR [18]. While the latter two are GAN-based, RRDB is trained using only L_1 loss. ProgFSR is a very recent SR method specifically designed for faces, shown to outperform several prior face SR approaches in [18]. It is trained on the full train split of CelebA, but using a bilinear kernel. For fair comparison, we therefore separately train and evaluate SRFlow on the same kernel.

Results are provided in Tab. 1 and Fig. 7. Since our aim is perceptual quality, we consider LPIPS the primary metric, as it has been shown to correlate much better with human opinions [27,54]. SRFlow achieves more than twice as good LPIPS distance compared to RRDB, at the cost of lower PSNR and SSIM scores. As seen in the visual comparisons in Fig. 7, RRDB generates extremely blurry

Table 2. General image SR results on the 100 validation images of the DIV2K dataset.

	DIV2K 4×							**DIV2K** 8×						
	PSNR↑	SSIM↑	LPIPS↓	LR-PSNR↑	NIQE↓	BRISQUE↓	PIQUE↓	PSNR↑	SSIM↑	LPIPS↓	LR-PSNR↑	NIQE↓	BRISQUE↓	PIQUE↓
Bicubic	26.70	0.77	0.409	38.70	5.20	53.8	86.6	23.74	0.63	0.584	37.16	6.65	60.3	97.6
EDSR [23]	28.98	0.83	0.270	54.89	4.46	43.3	77.5	-	-	-	-	-	-	-
RRDB [46]	29.44	0.84	0.253	49.20	5.08	52.4	86.7	25.50	0.70	0.419	45.43	4.35	42.4	79.1
RankSRGAN [55]	26.55	0.75	0.128	42.33	2.45	17.2	20.1	-	-	-	-	-	-	-
ESRGAN [46]	26.22	0.75	0.124	39.03	2.61	22.7	26.2	22.18	0.58	0.277	31.35	2.52	20.6	25.8
SRFlow $\tau = 0.9$	27.09	0.76	0.120	49.96	3.57	17.8	18.6	23.05	0.57	0.272	50.00	3.49	20.9	17.1

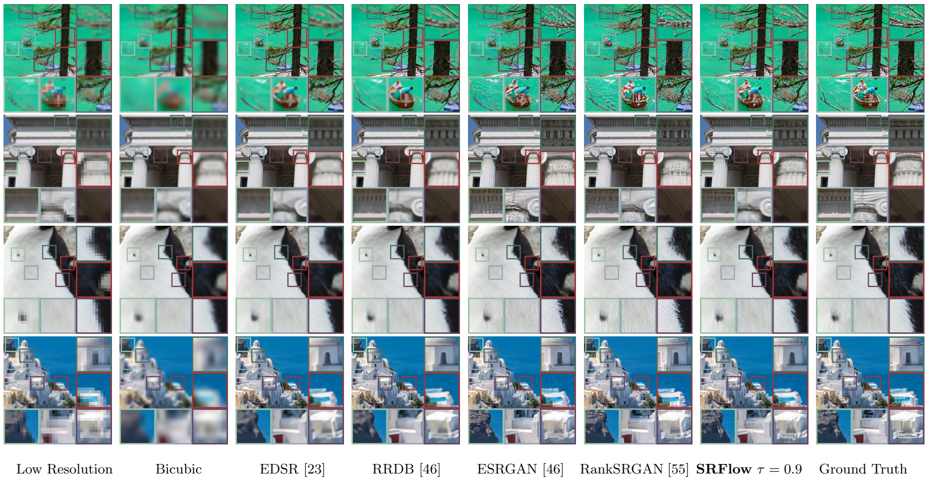
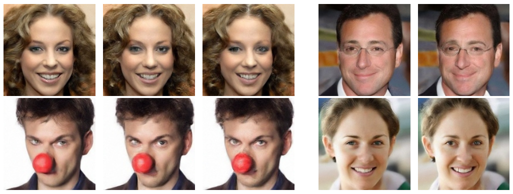


Fig. 8. Comparison to state-of-the-art for general SR on the DIV2K validation set.

results, lacking natural high-frequency details. Compared to the GAN-based methods, SRFlow achieves significantly better results in all reference metrics. Interestingly, even the PSNR is superior to ESRGAN and ProgFSR, showing that our approach preserves fidelity to the HR ground-truth, while achieving better perceptual quality. This is partially explained by the hallucination artifacts that often plague GAN-based approaches, as seen in Fig. 7. Our approach generate sharp and natural images, while avoiding such artifacts. Interestingly, our SRFlow achieves an LR-consistency that is even better than the fidelity-trained RRDB, while the GAN-based methods are comparatively in-consistent with the input LR image.

5.2 General Super-Resolution

Next, we evaluate our SRFlow for general SR on the DIV2K validation set. We compare SRFlow to bicubic, EDSR [23], RRDB [46], ESRGAN [46], and RankSRGAN [55]. Except for EDSR, which used DIV2K, all methods including SRFlow are trained on the train splits of DIV2K and Flickr2K (see Sec. 3.3). For the 4× setting, we employ the provided pre-trained models. Due to lacking availability, we trained RRDB and ESRGAN for 8× using the authors' code.



$K = 16$ Steps $K = 8$ Steps $K = 4$ Steps 196 Channels 64 Channels

Fig. 9. Analysis of number of flow steps and dimensionality in the conditional layers.

DIV2K 4×	PSNR↑	SSIM↑	LPIPS↓
No Lin. F-Step	26.96	0.759	0.125
No Affine Inj.	26.81	0.756	0.126
SRFlow	27.09	0.763	0.125

Table 3. Analysis of the impact of the transitional linear flow steps and the affine image injector.

EDSR and RRDB are trained using only reconstruction losses, thereby achieving inferior results in terms of the perceptual LPIPS metric (Tab. 2). Compared to the GAN-based methods [46,55], our SRFlow achieves significantly better PSNR, LPIPS and LR-PSNR and favorable results in terms of PIQUE and BRISQUE. Visualizations in Fig. 8 confirm the perceptually inferior results of EDSR and RRDB, which generate little high-frequency details. In contrast, SRFlow generates rich details, achieving favorable perceptual quality compared to ESRGAN. The first row, ESRGAN generates severe discolored artifacts and ringing patterns at several locations in the image. We find SRFlow to generate more stable and consistent results in these circumstances.

5.3 Ablative Study

To ablate the depth and width, we train our network with different number of flow-steps K and hidden layers in two conditional layers (4) and (5) respectively. Figure 9 shows results on the CelebA dataset. Decreasing the number of flow-steps K leads to more artifacts in complex structures, such as eyes. Similarly, a larger number of channels leads to better consistency in the reconstruction. In Tab. 3 we analyze architectural choices. The Affine Image Injector increases the fidelity, while preserving the perceptual quality. We also observe the transitional linear flow steps (Sec. 3.3) to be beneficial.

6 Conclusion

We propose a flow-based method for super-resolution, called SRFlow. Contrary to conventional methods, our approach learns the distribution of photo-realistic SR images given the input LR image. This explicitly accounts for the ill-posed nature of the SR problem and allows for the generation of diverse SR samples. Moreover, we develop techniques for image manipulation, exploiting the strong image posterior learned by SRFlow. In comprehensive experiments, our approach achieves improved results compared to state-of-the-art GAN-based approaches.

Acknowledgements: This work was supported by the ETH Zürich Fund (OK), a Huawei Technologies Oy (Finland) project, a Google GCP grant, an Amazon AWS grant, and an Nvidia GPU grant.

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: CVPR Workshops (2017)
2. Ahn, N., Kang, B., Sohn, K.A.: Image super-resolution via progressive cascading residual network. In: CVPR (2018)
3. Ardizzone, L., Lüth, C., Kruse, J., Rother, C., Köthe, U.: Guided image generation with conditional invertible neural networks. CoRR **abs/1907.02392** (2019), http://arxiv.org/abs/1907.02392
4. Bahat, Y., Michaeli, T.: Explorable super resolution. In: arXiv. vol. abs/1912.01839 (2019)
5. Behrmann, J., Grathwohl, W., Chen, R.T.Q., Duvenaud, D., Jacobsen, J.: Invertible residual networks. In: ICML. Proceedings of Machine Learning Research, vol. 97, pp. 573–582. PMLR (2019)
6. Bell-Kligler, S., Shocher, A., Irani, M.: Blind super-resolution kernel estimation using an internal-gan. In: NeurIPS. pp. 284–293 (2019), http://papers.nips.cc/paper/8321-blind-super-resolution-kernel-estimation-using-an-internal-gan
7. Bühler, M.C., Romero, A., Timofte, R.: Deepsee: Deep disentangled semantic explorative extreme super-resolution. arXiv preprint arXiv:2004.04433 (2020)
8. Dai, D., Timofte, R., Gool, L.V.: Jointly optimized regressors for image super-resolution. Comput. Graph. Forum **34**(2), 95–104 (2015). https://doi.org/10.1111/cgf.12544
9. Dinh, L., Krueger, D., Bengio, Y.: NICE: non-linear independent components estimation. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings (2015)
10. Dinh, L., Sohl-Dickstein, J., Bengio, S.: 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. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV. pp. 184–199 (2014). https://doi.org/10.1007/978-3-319-10593-2_13
12. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. TPAMI **38**(2), 295–307 (2016)
13. Durkan, C., Bekasov, A., Murray, I., Papamakarios, G.: Neural spline flows. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada. pp. 7509–7520 (2019)
14. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 2672–2680 (2014)
15. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: CVPR (2018)
16. Ignatov, A., Timofte, R., Van Vu, T., Luu, T.M., Pham, T.X., Van Nguyen, C., Kim, Y., Choi, J.S., Kim, M., Huang, J., et al.: Pirm challenge on perceptual image enhancement on smartphones: Report. arXiv preprint arXiv:1810.01641 (2018)
17. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. pp. 5967–5976 (2017).

https://doi.org/10.1109/CVPR.2017.632, https://doi.org/10.1109/CVPR.2017.632

18. Kim, D., Kim, M., Kwon, G., Kim, D.: Progressive face super-resolution via attention to facial landmark. In: arxiv. vol. abs/1908.08239 (2019)
19. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
20. Kingma, D.P., Dhariwal, P.: 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. pp. 10236–10245 (2018)
21. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR (2017)
22. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. CVPR (2017)
23. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. CVPR (2017)
24. Liu, R., Liu, Y., Gong, X., Wang, X., Li, H.: Conditional adversarial generative flow for controllable image synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 7992–8001 (2019)
25. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
26. Lugmayr, A., Danelljan, M., Timofte, R.: Unsupervised learning for real-world super-resolution. In: ICCVW. pp. 3408–3416. IEEE (2019)
27. Lugmayr, A., Danelljan, M., Timofte, R.: Ntire 2020 challenge on real-world image super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2020)
28. Lugmayr, A., Danelljan, M., Timofte, R., et al.: Aim 2019 challenge on real-world image super-resolution: Methods and results. In: ICCV Workshops (2019)
29. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR (2016), http://arxiv.org/abs/1511.05440
30. Menon, S., Damian, A., Hu, S., Ravi, N., Rudin, C.: Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In: CVPR (2020)
31. Mittal, A., Moorthy, A., Bovik, A.: Referenceless image spatial quality evaluation engine. In: 45th Asilomar Conference on Signals, Systems and Computers. vol. 38, pp. 53–54 (2011)
32. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a ”completely blind” image quality analyzer. IEEE Signal Process. Lett. **20**(3), 209–212 (2013)
33. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. The MIT Press (2012)
34. N., V., D., P., Bh., M.C., Channappayya, S.S., Medasani, S.S.: Blind image quality evaluation using perception based features. In: NCC. pp. 1–6. IEEE (2015)
35. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR. pp. 2536–2544. IEEE Computer Society (2016)
36. Pumarola, A., Popov, S., Moreno-Noguer, F., Ferrari, V.: C-flow: Conditional generative flow models for images and 3d point clouds. In: CVPR. pp. 7949–7958 (2020)

37. Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. pp. 1530–1538 (2015)
38. Sajjadi, M.S.M., Schölkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 4501–4510. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.481
39. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: ICCV. pp. 4570–4580 (2019)
40. Shocher, A., Cohen, N., Irani, M.: Zero-shot" super-resolution using deep internal learning. In: CVPR (2018)
41. Sun, L., Hays, J.: Super-resolution from internet-scale scene matching. In: ICCP (2012)
42. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L., Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., et al.: Ntire 2017 challenge on single image super-resolution: Methods and results. CVPR Workshops (2017)
43. Timofte, R., De Smet, V., Van Gool, L.: A+: Adjusted anchored neighborhood regression for fast super-resolution. In: ACCV. pp. 111–126. Springer (2014)
44. Timofte, R., Gu, S., Wu, J., Van Gool, L.: Ntire 2018 challenge on single image super-resolution: methods and results. In: CVPR Workshops (2018)
45. Timofte, R., Smet, V.D., Gool, L.V.: Anchored neighborhood regression for fast example-based super-resolution. In: ICCV. pp. 1920–1927 (2013). https://doi.org/10.1109/ICCV.2013.241, https://doi.org/10.1109/ICCV.2013.241
46. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C.C., Qiao, Y., Tang, X.: Esrgan: Enhanced super-resolution generative adversarial networks. ECCV (2018)
47. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Processing **13**(4), 600–612 (2004)
48. Winkler, C., Worrall, D.E., Hoogeboom, E., Welling, M.: Learning likelihoods with conditional normalizing flows. arxiv **abs/1912.00042** (2019), http://arxiv.org/abs/1912.00042
49. Yang, C., Yang, M.: Fast direct super-resolution by simple functions. In: ICCV. pp. 561–568 (2013). https://doi.org/10.1109/ICCV.2013.75, https://doi.org/10.1109/ICCV.2013.75
50. Yang, G., Huang, X., Hao, Z., Liu, M., Belongie, S.J., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. ICCV (2019)
51. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: CVPR (2008). https://doi.org/10.1109/CVPR.2008.4587647
52. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Processing **19**(11), 2861–2873 (2010). https://doi.org/10.1109/TIP.2010.2050625, https://doi.org/10.1109/TIP.2010.2050625
53. Yu, X., Porikli, F.: Ultra-resolving face images by discriminative generative networks. In: ECCV. pp. 318–333 (2016). https://doi.org/10.1007/978-3-319-46454-1_20
54. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. CVPR (2018)
55. Zhang, W., Liu, Y., Dong, C., Qiao, Y.: Ranksrgan: Generative adversarial networks with ranker for image super-resolution (2019)