Supplementary Material

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

1 Architecture Details

In this section, we give additional details about our SRFlow architecture. The construction of a flow-based architecture requires the flow layers to be invertible and have a tractable Jacobian log-determinant. Since super-resolution of diverse images has to be able to cope with different input sizes, we also ensure that our architecture is fully convolutional. We can therefore train our network on smaller patches, and directly apply it to the full image during testing. The computational time of our approach is 1.13 seconds for super-resolving one 256×256 input LR image with a scale factor of $4\times$ on an Nvidia V100 GPU.

1.1 Low-resolution Image Encoding

Our SRFlow network is conditioned on the encoding of the low-resolution image $\mathbf{u} = g_{\boldsymbol{\theta}}(\mathbf{x})$. To this end, we employ the RRDB-based architecture, described in the paper. It employs several RRDB-blocks with a channel dimension of 64, operating in the resolution of the input LR image. The final conditioning output $\mathbf{u} = g_{\boldsymbol{\theta}}(\mathbf{x})$ is achieved by concatenating the activations from 5 equally spaced RRDB blocks, resulting in a dimensionality of 320.

1.2 The Affine Injector Layer

Our affine injector layer provide a direct means of conditioning all dimensions of the flow feature-map $\mathbf{h}^n$ on the LR encoding as,

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

The scale and bias are extracted using non-invertible networks $f_{\boldsymbol{\theta},\mathrm{s}}(\mathbf{u})$ and $f_{\boldsymbol{\theta},\mathrm{b}}(\mathbf{u})$ respectively. The input $\mathbf{u}$ is first bilinearly resized to the resolution of the corresponding flow-level. A conv-ReLU block first reduces the dimensionality to 64. Another conv-ReLU block is then applied with 64-dimensional output. The output of $f_{\boldsymbol{\theta},\mathrm{s}}(\mathbf{u})$ and $f_{\boldsymbol{\theta},\mathrm{b}}(\mathbf{u})$ are then achieved by two separate conv-layers applied to the same 64-dimensional input. For these layers, we employ the zero-initialization strategy proposed in [4]. All convolutions have a 3×3 kernel.

1.3 Conditional Affine Coupling

This building block allows applying complex unconstrained conditional learned functions that act on the normalizing flow, without harming its invertibility. This is made possible by bypassing half of the activations and applying an affine transformation to the other half [2]. This transformation depends on the bypassed half $\mathbf{h}_A^n$ and conditional features $\mathbf{u}$ as,

$$\begin{cases} \mathbf{h}_A^{n+1} = \mathbf{h}_A^n \\ \mathbf{h}_B^{n+1} = \exp\left(f_{\boldsymbol{\theta},\mathrm{s}}^n(\mathbf{h}_A^n; \mathbf{u})\right) \cdot \mathbf{h}_B^n + f_{\boldsymbol{\theta},\mathrm{b}}^n(\mathbf{h}_A^n; \mathbf{u}) \end{cases} . \tag{2}$$

This expression can be easily inverted [2]. The network architectures of $f_{\boldsymbol{\theta},\mathrm{s}}$ and $f_{\boldsymbol{\theta},\mathrm{b}}$ are similar to those of the Affine Injector, described above. The only difference is that the two inputs $\mathbf{h}_A^n$ and $\mathbf{u}$ are initially concatenated after $\mathbf{u}$ is resized to the resolution of $\mathbf{h}_A^n$.

1.4 Squeeze Operation

This layer reshapes the activation map to half the width and height. In order to preserve the locality, neighboring pixels are stacked as seen in Figure 1.

1.5 Activation Norm

The Activation Norm (Actnorm) is a normalization layer. Unlike Batchnorm, it does not require synchronization among the elements of a batch. It simply consists of a learned scaling and bias factor for each dimension of the feature map. Thus it helps distributed learning on multiple GPUs.

2 Training Details

In this section, we give additional details about the training procedure for our SRFlow. We employ the Adam optimizer with a starting learning rate of $5 \cdot 10^{-4}$.

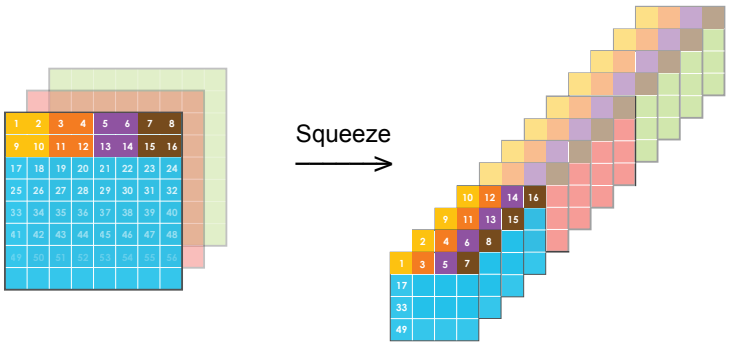


Fig. 1. Visualization of the Squeeze Operation.

$\tau = 0$ $\tau = 0.3$ $\tau = 0.6$ $\tau = 0.9$ Ground-Truth

Fig. 2. Super-resolved images sampled with different temperatures τ.

This learning rate is halved at $50\%, 75\%, 90\%$ and 95% of the total number of training iterations. During the first 50% of the training iterations, the pre-trained weights of the LR encoder $g_{\boldsymbol{\theta}}$ are frozen in a warm-up phase. In the latter 50%, all parameters of the SRFlow network, including $g_{\boldsymbol{\theta}}$, are optimized jointly with the same learning rate.

As has been observed in e.g. [4], adding slight random noise to the target image helps the training process and leads to better visual results. We therefore add Gaussian noise with a standard deviation of $\sigma = \frac{4}{\sqrt{3}}$ to the high-resolution image. In contrast to [4], we do not employ 5-bit quantization.

3 Detailed Quantitative Analysis

In this section, we provide additional quantitative analysis of our approach.

3.1 Influence of the Sampling Temperature

Here, we analyze the impact of the sampling temperature τ used during inference. It controls the variance of the Gaussian latent variable used when sampling SR images as $\mathbf{y} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z}; \mathbf{x})$, $\mathbf{z} \sim \mathcal{N}(0, \tau)$. As described in Section 4.1 of the main paper, a slightly reduced temperature $\tau < 1$, increases the image quality. When further decreasing the temperature to $\tau = 0$, the sampling process becomes deterministic. We analyze the effect of the sampling temperature τ on the main performance metrics, and on the sampling diversity itself. Results are shown in Figures 3, 4 and 5. A temperature $\tau = 0$ generates predictions with high fidelity, in terms of PSNR and SSIM. However, the results are blurry, as seen in Figure 2, explaining the poor perceptual quality (LPIPS) for this setting. Increasing the temperature leads to a drastic improvements in perceptual quality in terms of LPIPS distance. This is also clearly seen in the visual results in Figure 2. We also plot how the sampling diversity improves with increased temperature τ in terms of pixel-wise variance.

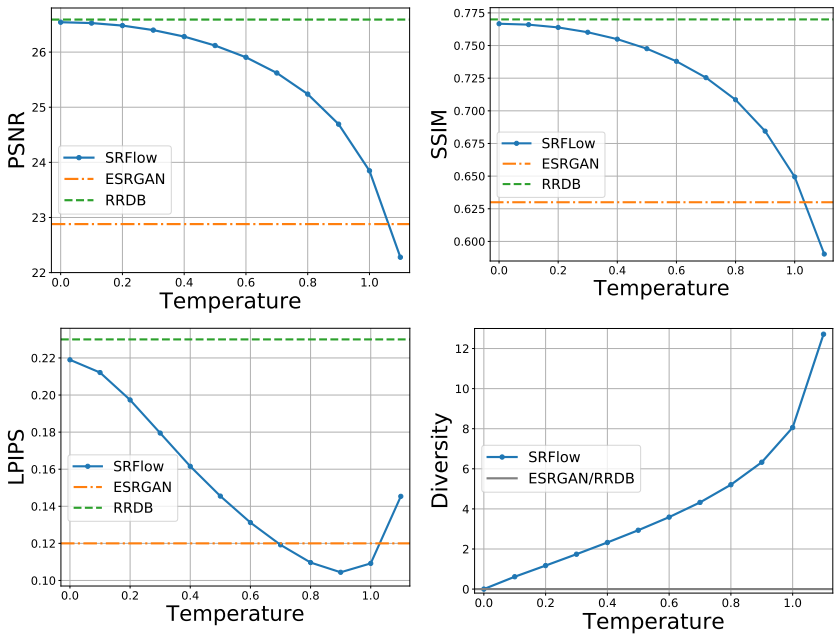


Fig. 3. Analysis of the sampling temperature τ in terms of PSNR, SSIM, LPIPS and sample diversity on CelebA (8×). Results of RRDB [7] and ESRGAN [7] are provided for reference.

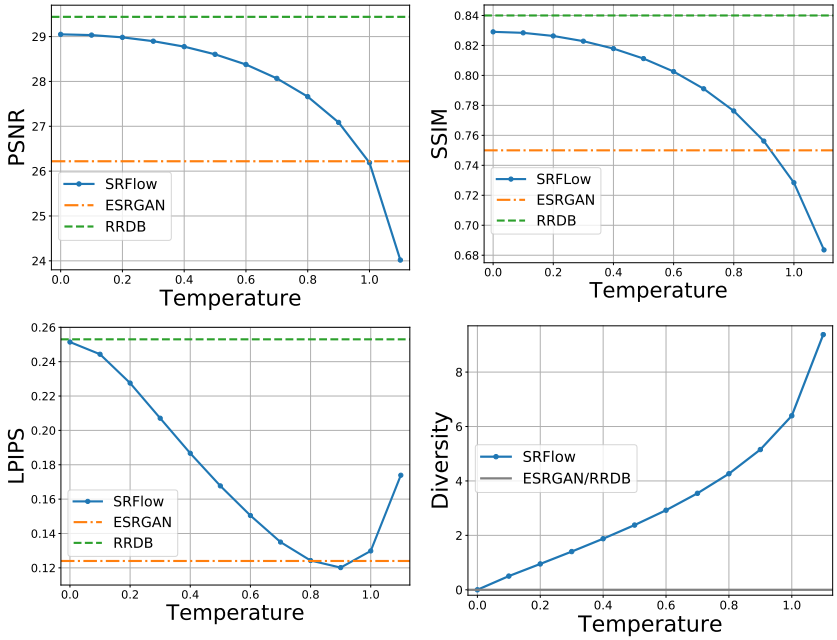


Fig. 4. Analysis of the sampling temperature τ in terms of PSNR, SSIM, LPIPS and sample diversity on DIV2K (4×). Results of RRDB [7] and ESRGAN [7] are provided for reference.

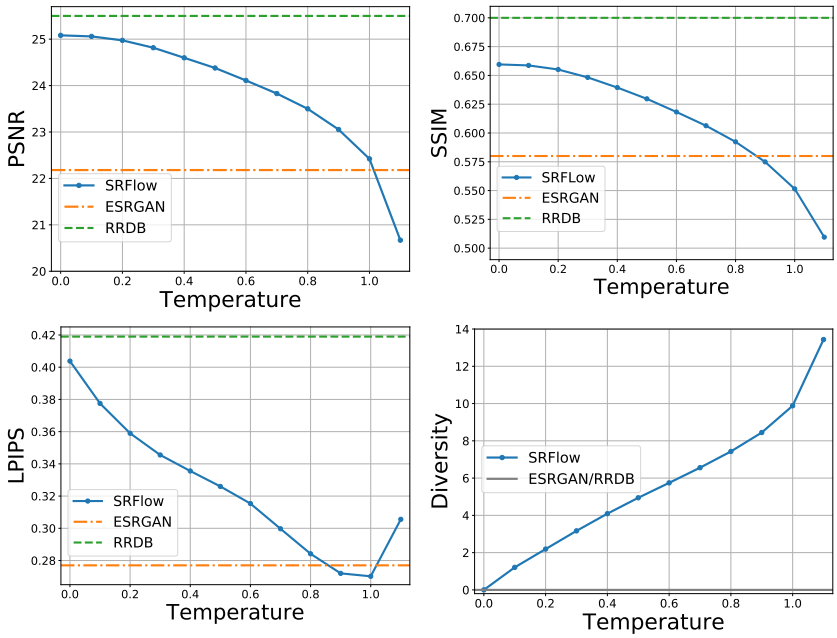


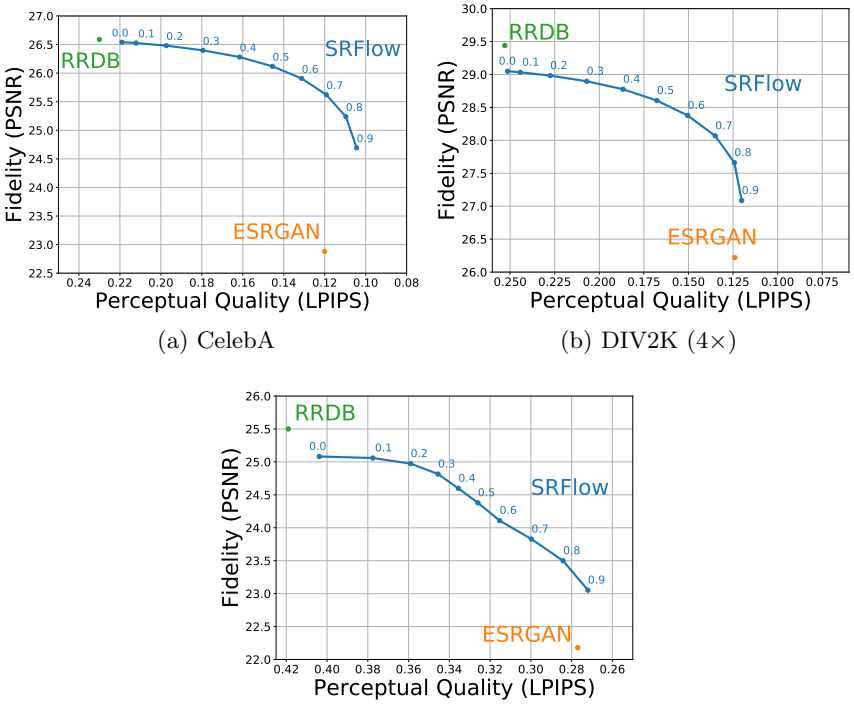
Fig. 5. Analysis of the sampling temperature τ in terms of PSNR, SSIM, LPIPS and sample diversity on DIV2K (8×). RRDB [7] and ESRGAN [7] are used as reference.

3.2 Perception–Distortion analysis

Here, we analyze the perception–distortion trade-off provided by our SRFlow. This trade off is an important choice decision for super-resolution methods [5,1]. While most techniques do not allow to influence the super-resolution process during inference, SRFlow provides an effective means of controlling this trade-off using the sampling temperature τ. We analyze this by plotting the perceptual quality (LPIPS) vs. the distortion (PSNR) with respect to the ground-truth in Figure 6. We plot the results for different τ for SRFlow. Our approach provides different alternative trade-offs. It achieves similar PSNR compared to the L_1-loss trained RRDB [7] for $\tau = 0$. On the other hand, SRFlow provides similar or better perceptual quality compared to ESRGAN [7] for $\tau \geq 0.8$, while achieving superior fidelity (PSNR).

3.3 Impact of LR-Encoder Initialization

To efficiently compare different variants of SRFlow, we reduced training time by pretraining the LR-Encoder g_{θ}. As shown in Table 1, the perceptual quality is comparable, while the fidelity is slightly higher, compared to using a randomly initalized LR-Encoder. The default SRFlow network was trained for 200k steps and uses a pretrained LR-Encoder, which was trained for 200k steps. The model without pretraining was trained for 300k iterations to make up for the missing pretraining. Since the main bottleneck during training is the calculation of the log determinant, this reduces training time.



(a) CelebA (b) DIV2K (4×)

(c) DIV2K (8×)

Fig. 6. Analysis of the trade-off between perceptual quality and fidelity (distortion). SRFlow allows the trade-off to be controlled by varying the sampling temperature τ. In comparison, RRDB [7] and ESRGAN [7] provide only a single operating point each.

Table 1. Quantitative comparison on CelebA between training the SRFlow model with and without first pretraining the LR-Encoder g_{θ}.

	PSNR	SSIM	LPIPS
Pretrained LR-Encoder	25.24	0.71	0.110
Without pretrained LR-Encoder	25.06	0.70	0.108

#Samples n	1	10	100	1000	10000	GT
LPIPS	0.108	0.105	0.099	0.098	0.093	0

Fig. 7. Best of n super-resolved ($8\times$) images in terms of the LPIPS metric.

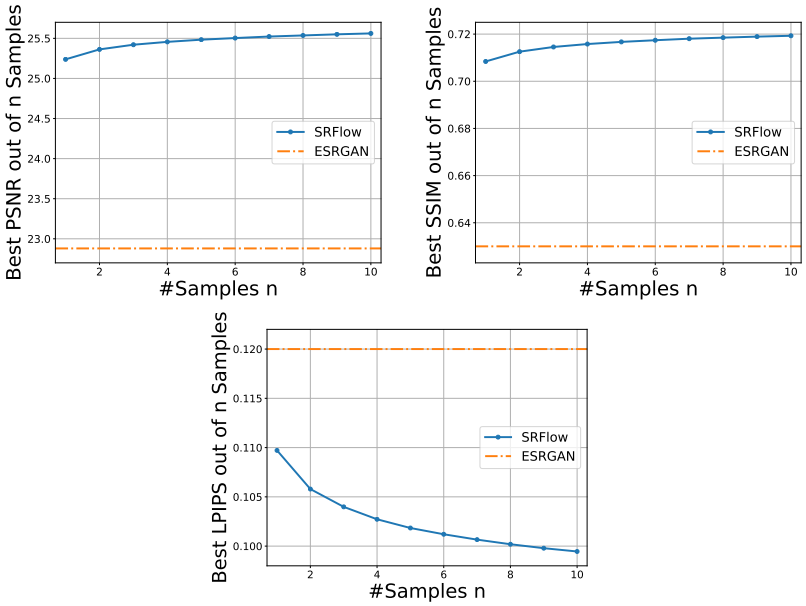


Fig. 8. Analysis of the improvement in performance metrics when choosing the best out of n samples. The performance of ESRGAN [7] is included for reference.

3.4 Oracle Analysis of the Sampling Space

As opposed to other state-of-the-art super-resolution approaches, SRFlow can be used to sample many variants of plausible super-resolutions. To further demonstrate the potential of this property, we analyze the performance of our SRFlow when selecting the best result among n random samples. Results, using a sampling temperature of $\tau = 0.8$, are shown in Figure 8. The results are computed over the full CelebA test set of 5000 images. The best result w.r.t. the ground-truth in each plot is selected based on the corresponding performance metric for $n = 1, \ldots, 10$ samples. This results shows that the perceptual quality in particular benefits from the oracle selection. This might be explained by our temperature setting, which forces the model to prefer perceptual quality over fidelity. It demonstrates that SRFlow provides a rich and diverse space of super-resolved images, from which solutions can be sampled. It provides the opportunity for improving the predictions of SRFlow by rejecting lower quality samples. A visual example is shown in Figure 7, when selecting the best out of n samples using the LPIPS distance.

Table 2. SRFlow results for image denoising on CelebA and DIV2K. Measurements for original images with Gaussian noise $\sigma = 20$, images that were super-resolved after downsampling, and restored images that use our latent space normalization approach, which also exploits the original HR image. We use the SRFlow model trained for 8× on CelebA and 4× on DIV2k

		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. 9. Image restoration examples on CelebA images with different degradations. Directly super-resolving (8×) the LR of the original removes noise but does not preserve details. Our SRFlow restoration also directly employs the original image by performing latent space normalization.

3.5 Image Restoration

We provide additional quantitative and qualitative results for image restoration, described in Section 4.5. Table 2 shows quantitative results for the task of image denoising when using white Gaussian noise with standard deviation $\sigma = 20$. We report performance metrics w.r.t. the clean ground-truth for the original noisy image, when just super-resolving the down-sampled image, and when using our restoration approach based on latent space normalization, as described in Section 4.5. Despite only being trained for the task of super-resolving clean images, our approach provides promising results for image denoising. This demonstrates the strong image posterior learned by our SRFlow. We show visual examples on CelebA and DIV2K in Figure 9 and Figure 10 respectively.

Fig. 10. Image denoising examples on DIV2k images. Directly super-resolving (4×) the LR of the original removes noise but does not preserve details. Our SRFlow restoration also directly employs the original image by performing latent space normalization.

4 Visual Results

In this section, we provide additional visual results.

4.1 State-of-the-Art for Face Super-Resolution

Additional examples that compare SRFlow with state-of-the-art for face super-resolution on CelebA are shown in Figure 11. For fair comparison, we also show SRFlow results when trained and applied on the same bilinear downsampling kernel as ProgFSR [3]. Our approach provides superior perceptual quality and better fidelity compared to the GAN-based methods.

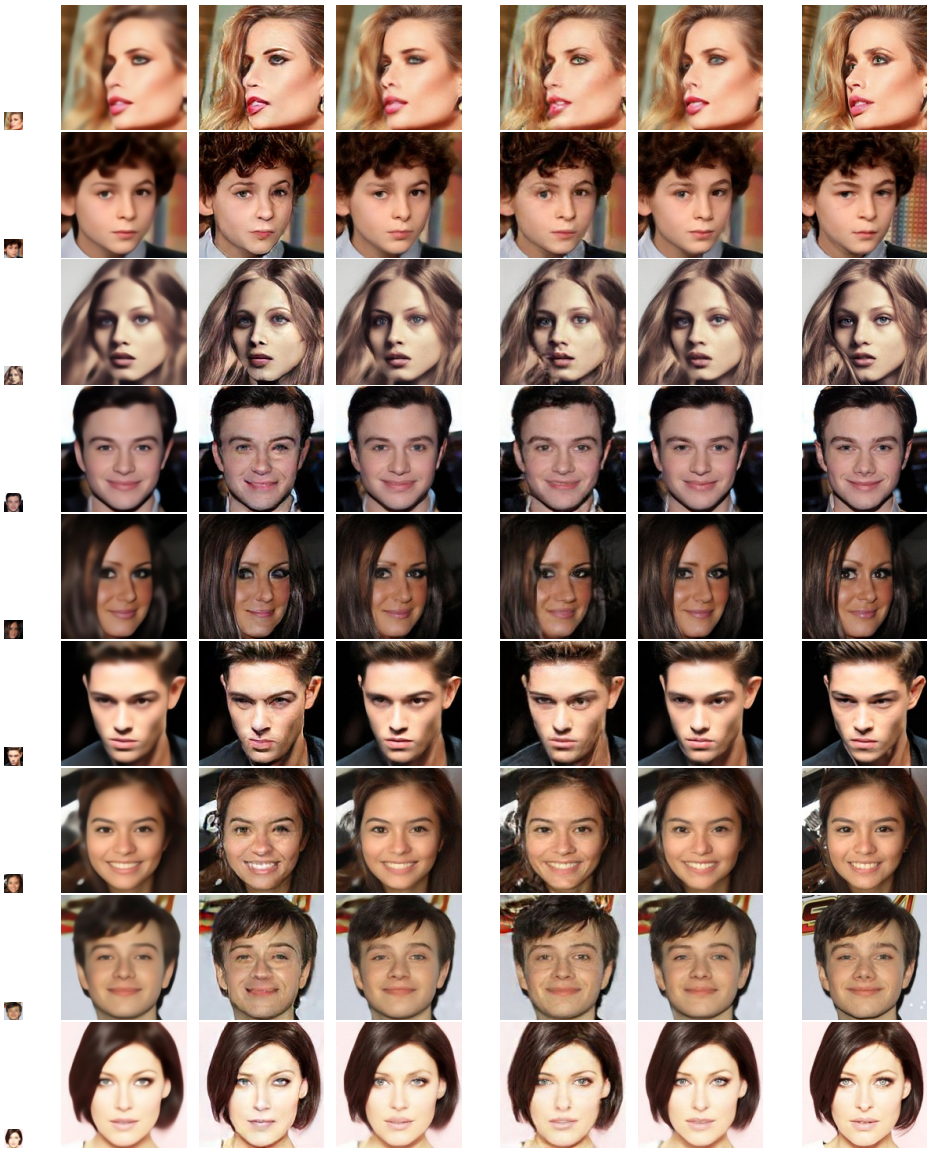


Fig. 11. Comparison of our SRFlow with state-of-the-art for 8× face super-resolution on CelebA. The three columns with super-resolutions on the left are trained and applied on bicubic downsampled images. The next two colums employ the bilinear kernel [3].

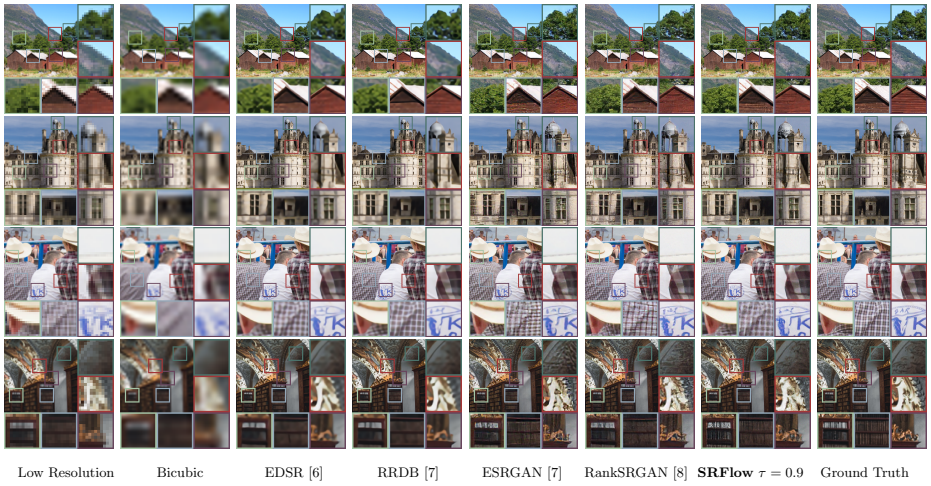


Fig. 12. Comparison to state-of-the-art for general super-resolution on the DIV2k 4× validation set.

4.2 State-of-the-Art General Super-Resolution

We provide more visual examples for the experiments on DIV2K, comparing SRFlow with with state-of-the-art super-resolution methods. In Figure 12 illustrates results for 4×. In addition, we provide results for DIV2K 8× in Figure 13. SRFlow achieves perceptual quality similar or better than ESRGAN in most cases. Moreover, our approach do not suffer from the hallucination artifacts typically seen in GAN-based methods.

4.3 Stochastic Face Super-Resolution

Here we provide additional examples to show the variety when sampling SR images with our default temperature $\tau = 0.8$ for CelebA. As seen for 8× super-resolution sampling in Figure 14, the low resolution image still contains significant information about facial characteristics. This bounds the diversity of super-resolution in order to be consistent. On the other hand in Figure 15 we show 16× super-resolution which is much more free while still being consistent to the low-resolution. Therefore one can observe a much higher variety.

4.4 Stochastic General Super-Resolution

In analogy to the visual sampling experiments for CelebA, we show results for the same procedure applied to DIV2K. An example for the variety of upscaling factor 4× is shown in Figure 16. For example, one can observe that the door in the lower right sometimes looks more like an archway and other examples more square. In addition we show the results for 8× upsampling in Figure 17. There it can be observed that the texture of the stones varies from being smooth to being rough.

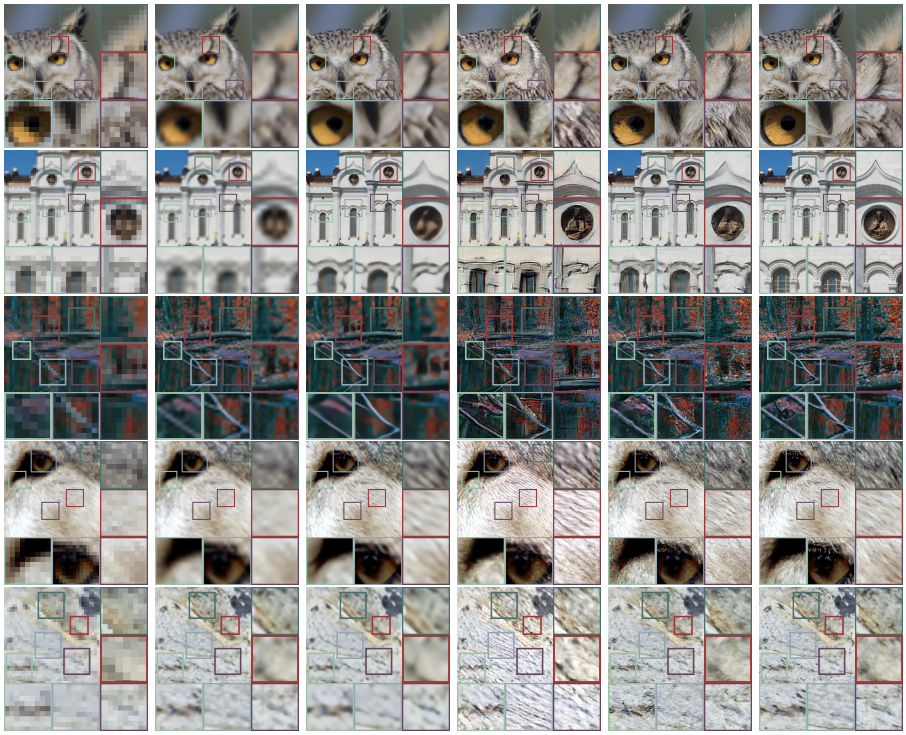


Fig. 13. Comparison to state-of-the-art for general super-resolution on the DIV2k 8× validation set.

4.5 Image Content Transfer

Additional examples for image content transfer are depicted in Figure 18. For this task we trained SRFlow with random shifts of 4px in HR to obtain a higher flexibility.

References

1. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: CVPR. pp. 6228–6237 (2018). https://doi.org/10.1109/CVPR.2018.00652, `http://openaccess.thecvf.com/content_cvpr_2018/html/Blau_The_Perception-Distortion_Tradeoff_CVPR_2018_paper.html`
2. 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)
3. Kim, D., Kim, M., Kwon, G., Kim, D.: Progressive face super-resolution via attention to facial landmark. In: arxiv. vol. abs/1908.08239 (2019)

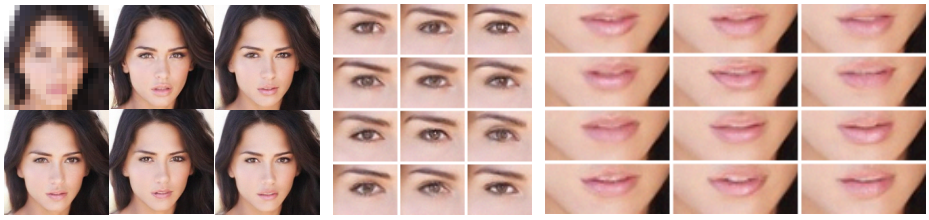


Fig. 14. Random SR samples generated by SRFlow using the given LR image on CelebA (8×).

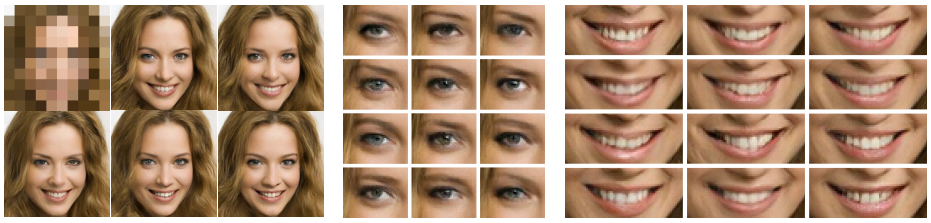


Fig. 15. Random SR samples generated by SRFlow using the given LR image on CelebA (16×).

Fig. 16. Random SR samples generated by SRFlow using the given LR image on DIV2K (4×).

Fig. 17. Random SR samples generated by SRFlow using the given LR image on DIV2K (8×).

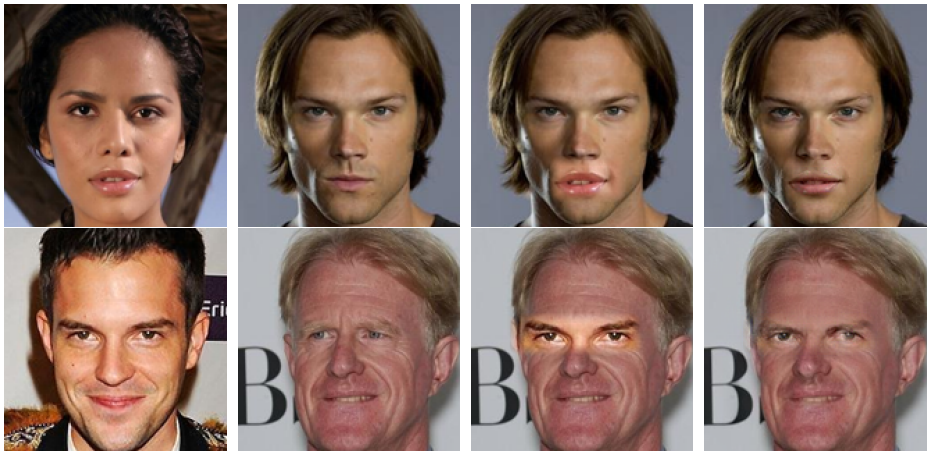


Fig. 18. Image content transfer for an existing HR image (top) and an SR prediction (bottom). Content from the source is applied directly to the target. By applying latent space normalization in our SRFlow, the content is integrated.

4. 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)
5. 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)
6. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. CVPR (2017)
7. 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)
8. Zhang, W., Liu, Y., Dong, C., Qiao, Y.: Ranksrgan: Generative adversarial networks with ranker for image super-resolution (2019)