Perceptual Losses for Real-Time Style Transfer and Super-Resolution: Supplementary Material

Justin Johnson, Alexandre Alahi, Li Fei-Fei
{jcjohns, alahi, feifeili}@cs.stanford.edu

Department of Computer Science, Stanford University

1 Network Architectures

Our style transfer networks use the architecture shown in Table 1 and our super-resolution networks use the architecture shown in Table 2. In these tables "$C \times H \times W$ conv" denotes a convolutional layer with C filters size $H \times W$ which is immediately followed by spatial batch normalization [1] and a ReLU nonlinearity.

Our residual blocks each contain two 3×3 convolutional layers with the same number of filters on both layer. We use the residual block design of Gross and Wilber [2] (shown in Figure 1), which differs from that of He *et al* [3] in that the ReLU nonlinearity following the addition is removed; this modified design was found in [2] to perform slightly better for image classification.

For style transfer, we found that standard zero-padded convolutions resulted in severe artifacts around the borders of the generated image. We therefore remove padding from the convolutions in residual blocks. A 3×3 convolution with no padding reduces the size of a feature map by 1 pixel on each side, so in this case the identity connection of the residual block performs a center crop on the input feature map. We also add spatial reflection padding to the beginning of the network so that the input and output of the network have the same size.

Layer	Activation size
Input	$3 \times 256 \times 256$
Reflection Padding (40×40)	$3 \times 336 \times 336$
$32 \times 9 \times 9$ conv, stride 1	$32 \times 336 \times 336$
$64 \times 3 \times 3$ conv, stride 2	$64 \times 168 \times 168$
$128 \times 3 \times 3$ conv, stride 2	$128 \times 84 \times 84$
Residual block, 128 filters	$128 \times 80 \times 80$
Residual block, 128 filters	$128 \times 76 \times 76$
Residual block, 128 filters	$128 \times 72 \times 72$
Residual block, 128 filters	$128 \times 68 \times 68$
Residual block, 128 filters	$128 \times 64 \times 64$
$64 \times 3 \times 3$ conv, stride 1/2	$64 \times 128 \times 128$
$32 \times 3 \times 3$ conv, stride 1/2	$32 \times 256 \times 256$
$3 \times 9 \times 9$ conv, stride 1	$3 \times 256 \times 256$

Table 1. Network architecture used for style transfer networks.

×4		×8	
Layer	**Activation size**	**Layer**	**Activation size**
Input	3 × 72 × 72	Input	3 × 36 × 36
64 × 9 × 9 conv, stride 1	64 × 72 × 72	64 × 9 × 9 conv, stride 1	64 × 36 × 36
Residual block, 64 filters	64 × 72 × 72	Residual block, 64 filters	64 × 36 × 36
Residual block, 64 filters	64 × 72 × 72	Residual block, 64 filters	64 × 36 × 36
Residual block, 64 filters	64 × 72 × 72	Residual block, 64 filters	64 × 36 × 36
Residual block, 64 filters	64 × 72 × 72	Residual block, 64 filters	64 × 36 × 36
64 × 3 × 3 conv, stride 1/2	64 × 144 × 144	64 × 3 × 3 conv, stride 1/2	64 × 72 × 72
64 × 3 × 3 conv, stride 1/2	64 × 288 × 288	64 × 3 × 3 conv, stride 1/2	64 × 144 × 144
3 × 9 × 9 conv, stride 1	3 × 288 × 288	64 × 3 × 3 conv, stride 1/2	64 × 288 × 288
-	-	3 × 9 × 9 conv, stride 1	3 × 288 × 288

Table 2. Network architectures used for ×4 and ×8 super-resolution.

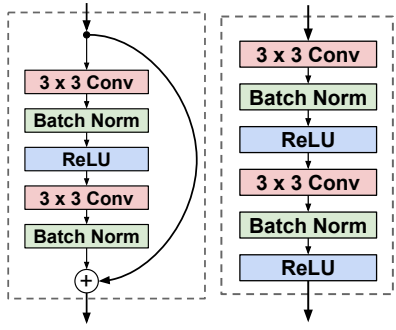


Fig. 1. Residual block used in our networks and an equivalent convolutional block.

2 Residual vs non-Residual Connections

We performed preliminary experiments comparing residual networks for style transfer with non-residual networks. We trained a style transfer network using *The Great Wave Off Kanagawa* as a style image, replacing each residual block in Table 1 with an equivalent non-residual block consisting of a pair of 3 × 3 convolutional layers with the same number of filters as shown in Figure 1.

Figure 2 shows the training losses for a residual and non-residual network, both trained using Adam [4] for 40,000 iterations with a learning rate of 1×10^{-3}. We see that the residual network trains faster, but that both networks eventually achieve similar training losses. Figure 2 also shows a style transfer example from the trained residual and non-residual networks; both learn similar to apply similar transformations to input images.

Our style transfer networks are only 16 layers deep, which is relatively shallow compared to the networks in [3]. We hypothesize that residual connections may be more crucial for training deeper networks.

Fig. 2. A comparison of residual vs non-residual networks for style transfer.

3 Super-Resolution Metrics

In Table 3 we show quantitative results for single-image super-resolution using the FSIM [5] and VIF [6] metrics.

		FSIM [5]				VIF [6]			
		Bicubic	ℓ_{pixel}	SRCNN [7]	ℓ_{feat}	Bicubic	ℓ_{pixel}	SRCNN [7]	ℓ_{feat}
×4	Set5 [8]	0.85	0.86	**0.89**	0.87	0.31	0.30	**0.38**	0.34
	Set14 [9]	0.85	0.85	**0.89**	0.88	0.26	0.24	**0.31**	0.28
	BSD100 [10]	0.76	0.76	0.80	**0.82**	0.22	0.21	**0.26**	0.24
×8	Set5 [8]	0.74	0.76	-	**0.79**	0.11	0.13	-	**0.15**
	Set14 [9]	0.72	0.74	-	**0.76**	0.09	0.11	-	**0.12**
	BSD100 [10]	0.63	0.64	-	**0.70**	0.08	0.09	-	**0.10**

Table 3. Quantitative results for super-resolution using FSIM [5] and VIF [6].

4 Super-Resolution User Study

In addition to using automated metrics, we performed a user study on Amazon Mechanical Turk to evaluate our ×4 super-resolution results on the BSD100 [10] dataset. In each trial a worker was shown a nearest-neighbor upsampling as well as the results from two different methods. Workers were told that we are "evaluating different methods for enhancing details in images" and were asked to "pick the enhanced version that you prefer". All trials were randomized, and five workers scored each image pair.

In Table 4 we show the results of the user study. For each pair of methods, we collected 5 votes for each of the 100 images in the BSD100 dataset. Table 4 shows both the raw number of votes cast for each method and the number of images for which a majority of users preferred one method over another. Between ℓ_{feat} and SRCNN, a majority of workers preferred the results of ℓ_{feat} on 96 / 100 images, and that between these two method workers cast 445 total votes for the results of ℓ_{feat} and just 55 votes for the results of SRCNN. These results support our claim that ℓ_{feat} results in visually pleasing super-resolution results.

	Majority Wins			Raw Votes		
	ℓ_{pixel}	SRCNN	ℓ_{feat}	ℓ_{pixel}	SRCNN	ℓ_{feat}
ℓ_{pixel}	-	0 / 100	0 / 100	-	14 / 486	21 / 479
SRCNN	100 / 0	-	4 / 96	486 / 14	-	55 / 445
ℓ_{feat}	100 / 0	96 / 4	100 / 0	479 / 21	445 / 55	-

Table 4. Results of the user study on Amazon Mechanical Turk comparing ×4 super-resolution results on the BSD100 dataset.

5 Super-Resolution Examples

We show additional examples of ×4 single-image super-resolution in Figure 4 and additional examples of ×8 single-image super-resolution in Figure 3.

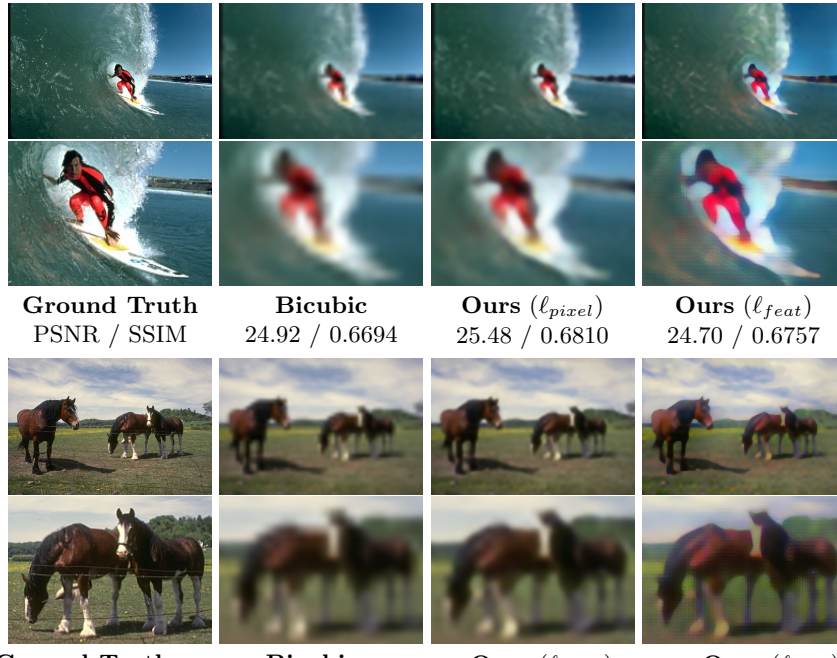
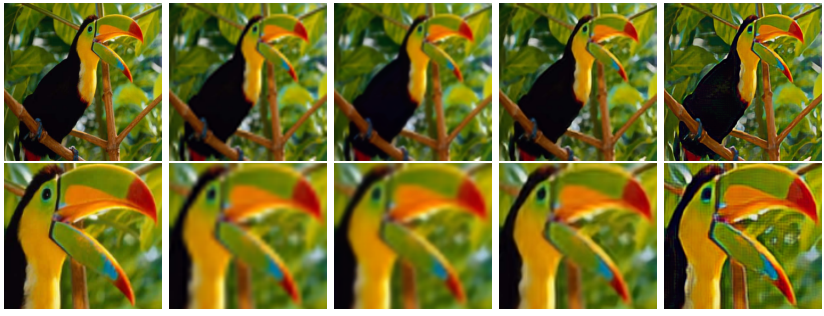
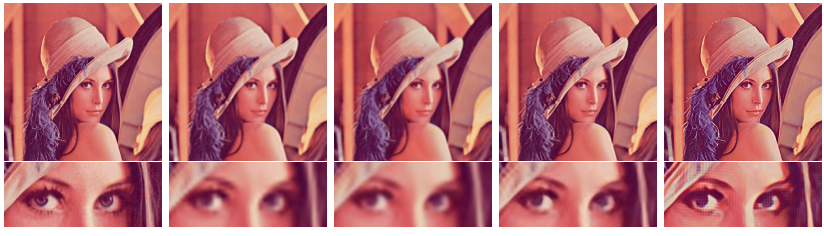


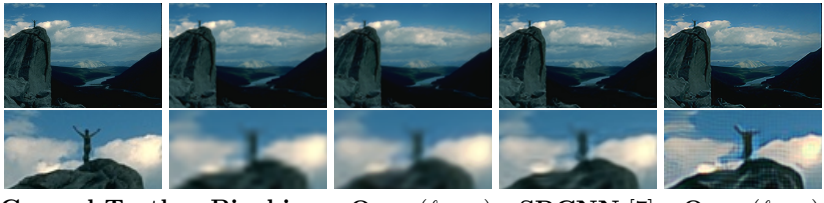
Fig. 3. Additional examples of ×8 single-image super-resolution on the BSD100 dataset.



Ground Truth	**Bicubic**	**Ours** (ℓ_{pixel})	**SRCNN** [7]	**Ours** (ℓ_{feat})
PSNR / SSIM	30.18 / 0.8737	29.96 / 0.8760	32.00 / 0.9026	27.80 / 0.8053



Ground Truth	**Bicubic**	**Ours** (ℓ_{pixel})	**SRCNN** [7]	**Ours** (ℓ_{feat})
PSNR / SSIM	29.84 / 0.8144	29.69 / 0.8113	31.20 / 0.8394	28.18 / 0.7757



Ground Truth	**Bicubic**	**Ours** (ℓ_{pixel})	**SRCNN** [7]	**Ours** (ℓ_{feat})
PSNR / SSIM	32.48 / 0.8575	32.30 / 0.8568	33.49 / 0.8741	30.85 / 0.8125

Fig. 4. Additional examples of $\times 4$ single-image super-resolution on examples from the Set5 (top), Set14 (middle) and BSD100 (bottom) datasets.

References

1. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. (2015)
2. Gross, S., Wilber, M.: Training and investigating residual nets. http://torch.ch/blog/2016/02/04/resnets.html (2016)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
4. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2015)
5. Zhang, L., Zhang, L., Mou, X., Zhang, D.: Fsim: a feature similarity index for image quality assessment. IEEE transactions on Image Processing **20**(8) (2011) 2378–2386
6. Sheikh, H.R., Bovik, A.C.: Image information and visual quality. IEEE Transactions on Image Processing **15**(2) (2006) 430–444
7. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV. (2014)
8. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. (2012)
9. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces. Springer (2010) 711–730
10. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. (2015)