New Techniques for Preserving Global Structure and Denoising with Low Information Loss in Single-Image Super-Resolution

Yijie Bei Alex Damian Shijia Hu Sachit Menon Nikhil Ravi Cynthia Rudin*
Duke University

Abstract

This work identifies and addresses two important technical challenges in single-image super-resolution: (1) how to upsample an image without magnifying noise and (2) how to preserve large scale structure when upsampling. We summarize the techniques we developed for our second place entry in Track 1 (Bicubic Downsampling), seventh place entry in Track 2 (Realistic Adverse Conditions), and seventh place entry in Track 3 (Realistic difficult) in the 2018 NTIRE Super-Resolution Challenge. Furthermore, we present new neural network architectures that specifically address the two challenges listed above: denoising and preservation of large-scale structure.

1. Introduction

Super-resolution (SR) is a classic problem in image processing where the goal is to generate a high resolution image from one or more low resolution images. Applications of super-resolution are wide-ranging. For instance, SR is important for allowing modern high-definition displays to function properly when showing video recorded at lower resolutions. SR also has many applications in medical imaging, such as reducing noise in images stemming from uncontrollable patient motions (11). This work focuses on single image super-resolution, which is useful for photographic enhancement, license plate recognition, satellite imaging, and other remote sensing applications such as recognition of a military target (16).

Deep learning techniques can learn a mapping directly from low resolution to high resolution images, where all feature construction is automated. This makes some types of complex preprocessing much easier than previous approaches, for example, we no longer need to explicitly choose a dictionary of low-level features (e.g., edge detectors) to convolve with the image. The fact that training deep neural networks has become much easier within the past few years has led to more reliable automated training. On the other hand, the fact that these deep learning methods use recursive mathematical formulas that are now much more complicated than before makes it more difficult to determine how to best troubleshoot them to achieve higher-quality performance.

In this work we discuss several insights into the problem of single-image super-resolution – many of which have led to higher quality performance beyond entries from last year's NTIRE single-image SR competition. These insights concern the amplification of noise when upsampling and the preservation of large scale structure in enhanced images. We introduce neural network architectures for both the denoising problem (DeNoising for Super-Resolution – DNSR) and the problem of preserving large-scales structure (Automated Decomposition and Reconstruction for Super-Resolution – ADRSR). Additionally we present a set of tricks that provided boosts in SR performance.

For denoising while upsampling, we present the DNSR (and more basic DNISR) architecture that concatenates two networks, where the first network is for denoising and the second is a baseline method for SR. This leverages domain knowledge that the noise should not have been in the low-resolution image in the first place and thus we should not amplify it. Training these concatenated networks led to improvements in performance in Track 2 (realistic mild adverse conditions) and Track 3 (realistic difficult) of the NTIRE SR 2018 challenge.

Modern methods for SR have trouble preserving large scale structure. Even if the high resolution images look realistic in local patches, the global structure (such as stripes that reach across the full image) can have serious visible faults. We present an architecture for preserving structure at multiple scales. In our network, ADRSR, the original image is downsampled multiple times, convolutions are performed on each of the downsampled images, and combined to form the final high-resolution image. This directly provides the network with more information about the larger scales (that it thus does not need to learn).

The architectures for denoising and preserving large-scale structure can be used with any network blocks used

*All authors contributed equally. Thanks to other members of Duke Data Science Team

for SR; we used convolutional blocks from EDSR (9) within our implementations, but these can be changed to any other blocks. DNISR or DNSR combine any network for denoising with any network for SR.

Most of the ideas discussed here were not implemented in time for the NTIRE 2018 SR competition deadline. However, we present a set of tricks that were helpful in achieving higher level performance during the competition. For instance, an idea used in our Track 1 (classic bicubic downsampling) entry was to randomly shuffle the red, green, and blue layers of the image during training, which helps as a form of self-ensembling. We also discuss different upsampling techniques, and find that for x8 amplification, we should learn the fully amplified image directly, because learning a x4 followed by a x2 amplification tend to lead to the spurious addition of details that do not exist in the original high-resolution image.

All of these ideas were developed over the course of approximately 8 weeks by a team of 5 undergraduates with no previous experience in image processing.

Our entries in the 2018 NTIRE superresolution competition (13) achieved seventh place in Track 2 (realistic mild adverse conditions), seventh place in Track 3 (realistic difficult) and second place in Track 1 (classic bicubic downsampling).

	Track 1	Track 2	Track 3
PSNR	25.433	23.374	21.658
SSIM	0.7067	0.6252	0.5400

Table 1: Competition Result

2. Previous Work

Many approaches to single-image super-resolution are based on different methods of image upsampling. In particular, nearest-neighbors upsampling (in which each unknown pixel in the upsampled image is assigned the value of its nearest known neighbor) and bicubic upsampling (in which each unknown pixel in the upsampled image is assigned a value interpolated from its nearest known neighbors) are popular methods for basic upsampling (2; 3). These methods, while simple and computationally efficient, do not provide realistic high-resolution images. More advanced methods attempt to build a map between low resolution images and high resolution images through a variety of different techniques. Some techniques include frequency-domain methods such as alias removal (14), recursive least squares (7), and multichannel sampling theorem methods (15), as well as spatial-domain methods, such as iterated back-projection (6), joint MAP restoration (4), and adaptive filtering (10).

Neural networks have recently been successful for image processing tasks, and through application of classical ResNet architectures, Ledig et al. created one successful example of a convolutional neural network for super-resolution, called SRResNet (8). Their work showed that the use of residual blocks improved performance on super-resolution tasks over more traditional convolutional neural network architectures, and has become the basis for many future architectures for super-resolution. Lim et al. then improved on this with their EDSR method by removing batch normalization, using an L1 rather than L2 loss function, and adding depth to the network (9). While these models have seen some success in the super-resolution task for 'clean' images (that is, images that have been bicubically downscaled with no further degradations), they do not show good results for images with noise, blur, or other degradations.

A few recent interesting super-resolution techniques have been suggested for degraded images. Zhang et al. (18) suggested using CNN denoisers as a modular part of model-based optimization methods to perform various computer vision tasks including super resolution. Shocher et al. (12) proposed an unsupervised approach that trains an image-specific CNN at test time that learns to use the repetitive structure of images to fill in details where there previously were none.

Other neural-network based methods, such as generative adversarial networks (8), have shown success in super-resolution as measured by human viewers. However, these networks achieve visual effects suitable for human viewing by 'hallucinating' features from the low resolution image that are not necessarily in the original image, but would be believable given the low resolution image. As such, they are not as well suited for tasks that maximize similarity to the original high resolution image, such as PSNR and SSIM.

The methods introduced into this work are different in that they heavily leverage prior knowledge: DNSR leverages the knowledge that denoising before upsampling is helpful, while ADRSR uses a pyramid of downsampled images to borrow information at broader scales. The ideas within ADRSR and DNSR can be combined with any neural network approaches to denoising and super-resolution in order to include domain knowledge.

3. Challenges

When approaching all three super-resolution tracks (corresponding to non-noisy and noisy images), we encountered multiple challenges.

First, there were challenges that were specific to the competition itself. One such challenge was that of *model validation*, because the PSNR values of our algorithm varied wildly between images (see Figure 1). Depending on which 100-image subset we used for validation, average PSNR values ranged from 22 to 27. This made it difficult

to compare our results to others' and required us to fix a validation set of 100 images throughout training.

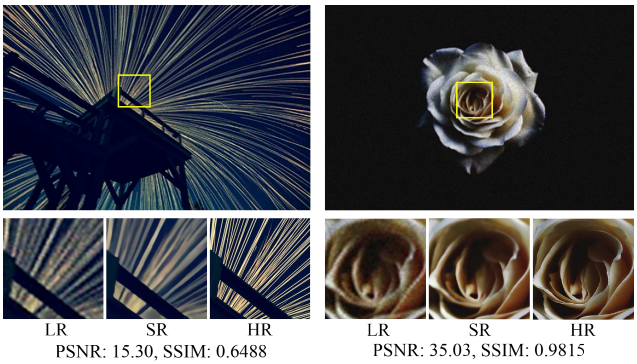

Figure 1: Varying PSNR between our algorithm's images for Track 2

Particularly for noisy images, it is very difficult to avoid *amplifying the noise while upsampling*. Several of the techniques we introduce here were useful for this, particularly the denoising and upsampling network DNSR for Tracks 2 and 3. Even without noise, artifacts tend to appear when upsampling by a factor of eight.

Most traditional denoisers require some knowledge of the noise itself, normally the standard deviation. To use any of these denoisers, it was imperative to *reverse engineer* the noise. We took approximately flat areas of various images and considered the difference between the degraded low resolution images and down-scaled versions of the high resolution images. Because a blur kernel has no effect on flat regions of an image, this difference should be a good approximation of the noise (see Figure 2).

Most prior convolutional networks for super-resolution tend to focus on increasing the resolution in local areas; however, this approach does not *take into account more global patterns* (such as zebra stripes). Some recent work (5; 12) have aimed to solve this problem in other promising ways, and we present a new method for handling this (ADRSR) in what follows.

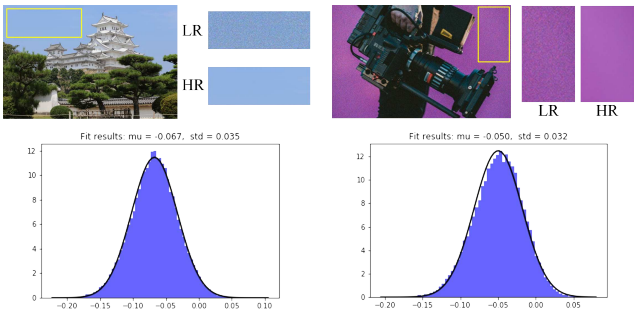

Figure 2: Histogram of noise from two images

4. ADRSR: A type of architecture that preserves global structure

Figure 8 shows the types of problems that can arise from EDSR and similar SR algorithms. These algorithms consider local image patches, and do not aim to reconcile them with larger-scale patterns that crosscut into different patches. Both increasing the depth of the network and increasing the size of each kernel allows the network to include larger scale patterns. However, these approaches are either hard to train, or do not converge at all. Thus, we reasoned that these larger patterns could be detected even by using a smaller kernel on a downsampled image without significant loss of information; the flexibility afforded by a large number of larger kernels may be unnecessary to capture this information.

The architecture that we introduce for preserving global structure is presented in Figure 3, called Automated Decomposition and Reconstruction for SR (ADRSR). The original image is downsampled several times, with each downsampled image being fed through a parallel super-resolution network. This pyramid representation for the input allows us to create filters that capture patterns from the original image at various scales. We then iteratively combine the information from the various upscaled images to produce a final, more accurate image that respects global structure. When running the network forward on a new image, it would start from the coarsest scale, and iteratively add more detail on the finer scales.

In Figure 3, the SR network labeled in the figure can be replaced with any SR network.

5. DNSR: A type of architecture for denoising with low information loss for SR

As the principal challenge for Tracks 2 and 3 is noise, we considered three possible approaches for dealing with the noise:

- (Baseline simple approach). The simplest approach is to manually preprocess the images with a noise reduction algorithm, and then train a super resolution convolutional network on the denoised images.
- (SR without denoising). Allow the residual blocks in a super-resolution network (such as EDSR) to simultaneously denoise the input images and extract features. That is, we directly train EDSR on the noisy data.
- (DNISR, DNSR) After training the denoising network and SR network separately, we concatenate them, and then continue to train them together as a single network. DNISR and DNSR differ in the way that they concatenate the two networks during the final training stage, see Figure 4.

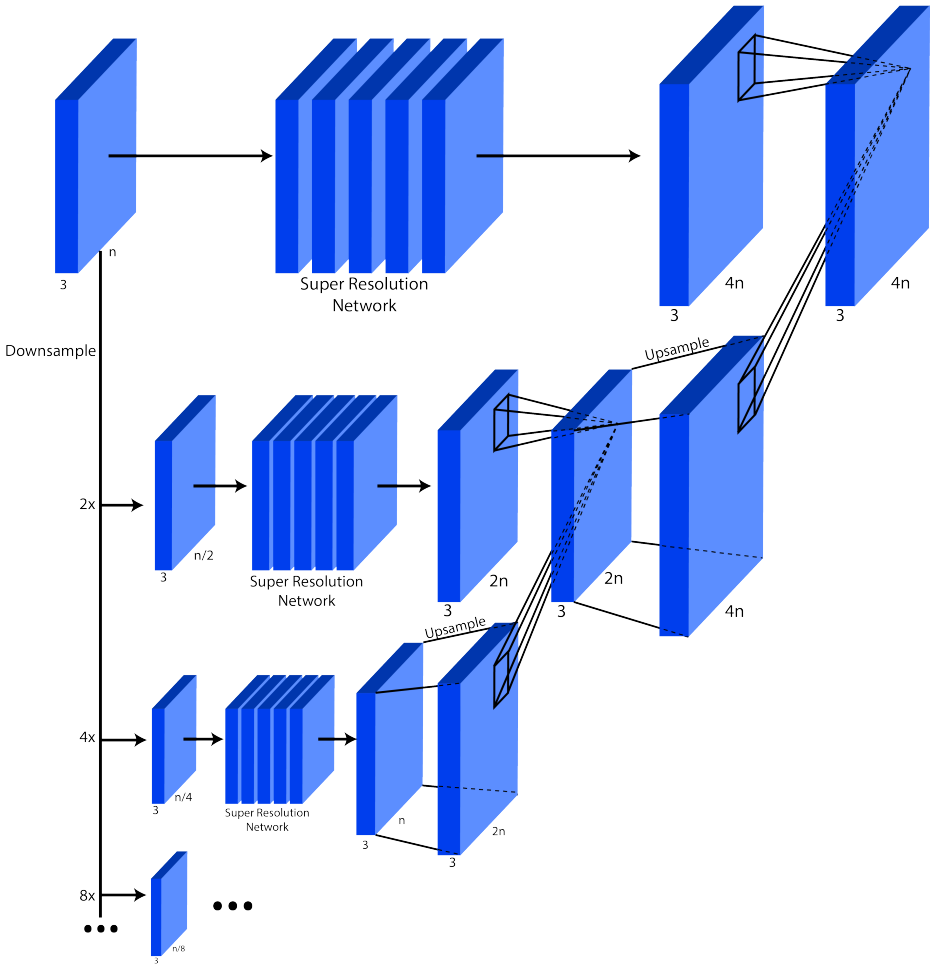

Figure 3: ADRSR Network with a x4 super resolution network

The first baseline approach allowed us to incorporate domain knowledge about the noise, but performed poorly due to the information loss caused by the denoiser. The second approach, on the other hand, did not tend to suffer from information loss. However, it was not possible to incorporate any domain knowledge about the problem (for instance that the image needs to be denoised) into the network. The third and fourth approaches solved both problems. They allowed us to incorporate domain knowledge into the network, since we could explicitly train the denoising network. DNSR trains the denoiser and super-resolution network together at the end to minimize information loss of the overall procedure. This approach is also advantageous when given a small number of images with the same degradations applied. After reverse-engineering the noise, external data can be used to train the denoising and super-resolution networks, and then the entire concatenated network can be trained on the dataset to allow the network to correct any additional degradations.

Based on our final approach, we constructed two models, which perform the concatenation in two different ways. The first was DNISR (DeNoising Into Super-Resolution), which ran the image through a denoising network (we used DNCNN (17)), producing a low-resolution noise-reduced image, and then ran the result through the super-resolution network (we used EDSR) to produce a high-resolution image.

We found a useful trick to further minimize information loss in DNISR: we fed the original image into the super-resolution network alongside the noise-free image with weights initialized to 0.

The second approach (DeNoising and Super-Resolution – DNSR) used a more complicated concatenation procedure. It removed the information bottleneck between the two networks by combining the tail layer of the denoiser (which mapped 256 channels to 3) and the head layer of the SR network (which mapped 3 channels to 256) into a single bridge convolutional layer that mapped directly from

the number of feature maps in the denoiser to the number of feature maps in the SR network. Unlike DNISR, there is no denoised image produced before entering the super-resolution network. See Figure 4 for the architecture. We submitted the same model for Tracks 2 and 3 of the NTIRE 2018 competition.

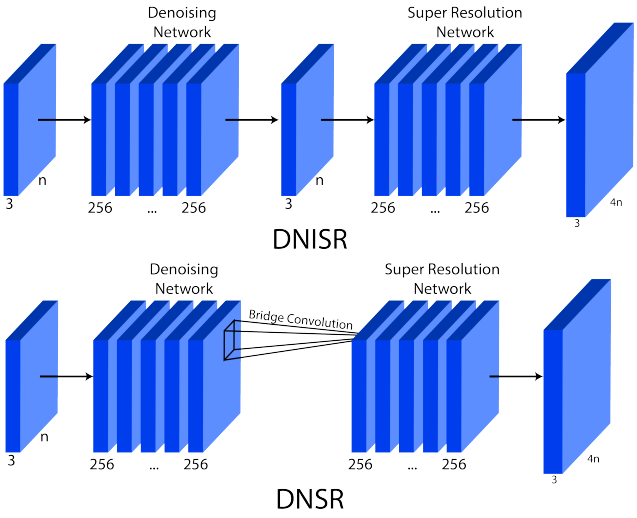

Figure 4: Architectures for DNSR and DNISR.

Table 2 and Figures 5 and 6 show a PSNR comparison for EDSR, DNISR, and DNSR. For a visual comparison of the images produced by each algorithm, see Figure 12.

Algorithm	BICUBIC	EDSR	DNISR	DNSR
PNSR	23.47	24.49	24.52	24.90
SSIM	0.7333	0.7925	0.7940	0.7956

Table 2: Comparison of results from EDSR and our denoising networks. The numbers reported were computed on the DIV2K (1) validation data set.

6. General Tricks and Insights

We discovered several tricks that can be used any time, with almost any network architecture. To see the results these tricks had on upscaling images by a factor of 8, see Figure 12.

- RGB Layer Shuffle: In addition to flipping and rotating the image patches during training and generation, we randomly shuffled the red, green, and blue layers. This improved our overall model by a small amount. This trick is applicable to any convolutional structure. Figure 11 shows the effect of test-time RGB Shuffling.
- Per-Image Mean Shift: Instead of calculating the average mean throughout all of the images and normalizing by that value, as in the original EDSR paper, we instead normalized each individual image patch during training by subtracting its mean.

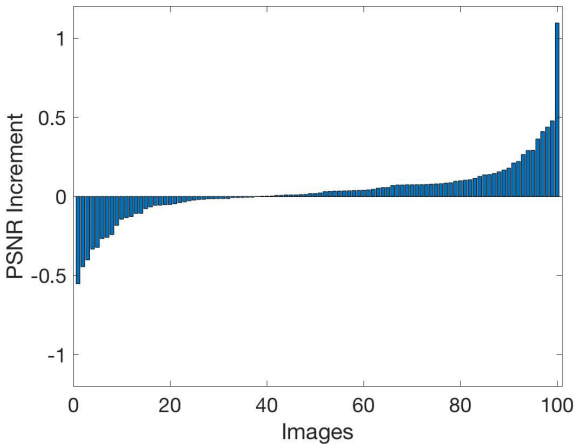

Figure 5: PSNR difference between DNISR and EDSR (sorted by difference in PSNR) on the 100 image validation set from DIV2K (1).

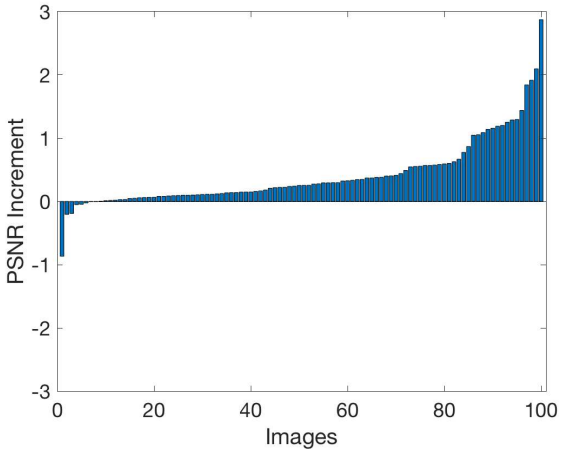

Figure 6: PSNR difference between DNSR and EDSR (sorted by difference in PSNR) on the 100 image validation set from DIV2K (1)

- Different Upsampling Techniques: For Track 1, we started by using sub-pixel shift to upscale the image. In addition, to upsample by a factor of 8, we concatenated three $\times 2$ upsamplers, as in the original EDSR paper. Using this approach, we ran into artifacts induced by the upscaling (see Figure 7). These artifacts were diminished by switching the upsampling method to Transposed Convolution upsampling. However, even with the sub-pixel shift upscaler, the problem went away when we switched to directly learning a $\times 8$ upscaler instead of three concatenated $\times 2$ up-

scalers.

In our final method, we found that direct $\times 8$ upscaling combined with the sub-pixel shift upscaler produced images with higher PSNR values. However, the concatenated $\times 2$ upscalers seemed less prone to creating artifacts due to antialiasing (see Figure 8).

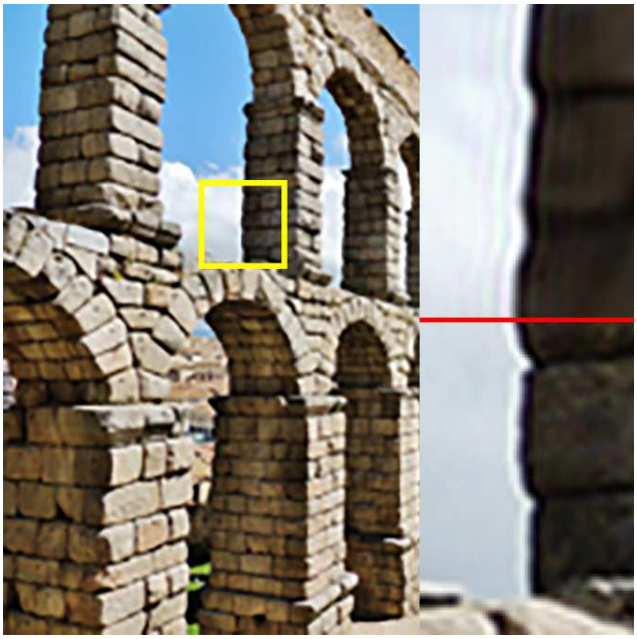

Figure 7: The upscaler using sub-pixel shift (top-right) has clear chromatic artifacts, while the upscaler using transposed convolutional upscaling (bottom-right) does not.

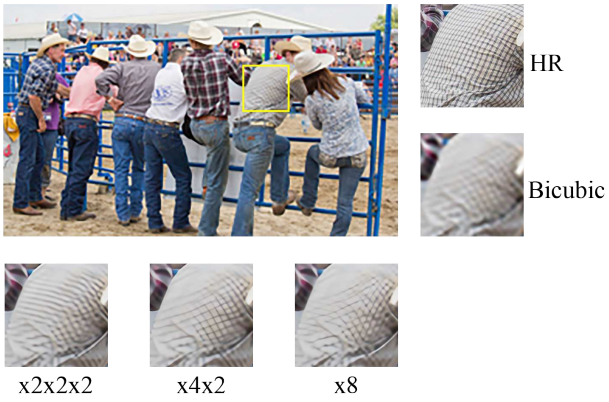

Figure 8: Diagonal lines created by some upscaling methods due to anti-aliasing

- Residual Scaling Factor: In EDSR, each residual layer is multiplied by 0.1 at the end. Instead of hardcoding this parameter, we allowed it to be a free variable that could be trained.

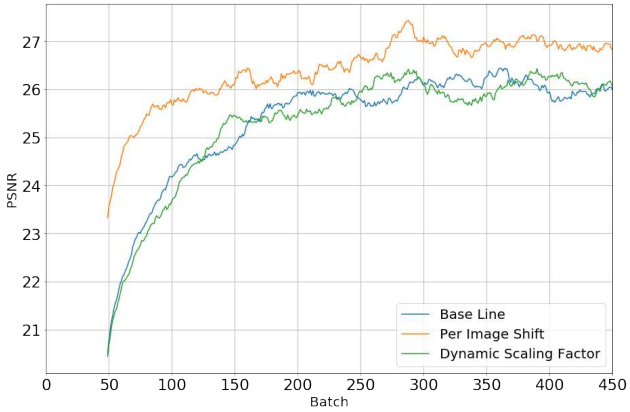

Figure 9: Convergence rates for baseline EDSR, EDSR with per image shift and dynamic scaling factor. The figure is plotted with rolling mean of 50.

- Edge Loss: We attempted to add an edge-loss component to the loss by applying a Sobel filter to both the upscaled and ground-truth images, and comparing those. However, this did not improve on our previous model.

- Kernel Size: We tried various kernel sizes, however 2×2 produced worse results and we could not successfully train the network with the 4×4 and 5×5 kernel sizes.

Figure 9 shows a comparison of convergence rates for baseline EDSR, EDSR with per image intensity shift, and EDSR with dynamic residual scaling factors. All training started with randomly initialized weights. Using per-image mean shift gives a higher initial PSNR and faster convergence.

Table 3 and Figure 10 show a PSNR comparison for VDSR, EDSR, and our improved EDSR model. For a visual comparison of the images produced by each algorithm, see Figure 12.

Algorithm	EDSR	Improved EDSR	VDSR	BICUBIC
PNSR	25.49	25.60	24.70	23.69
SSIM	0.6930	0.6974	0.6580	0.6291

Table 3: Comparison of our improved EDSR algorithm to several baselines. The improvements could have been made to any SR algorithm besides EDSR. The numbers reported were computed on the DIV2K validation data set.

Figure 11 shows the PSNR increment across the 100 images from the DIV2K validation set after applying only RGB Shuffling to EDSR. In 97 out of the 100 cases, there was a boost in PSNR.

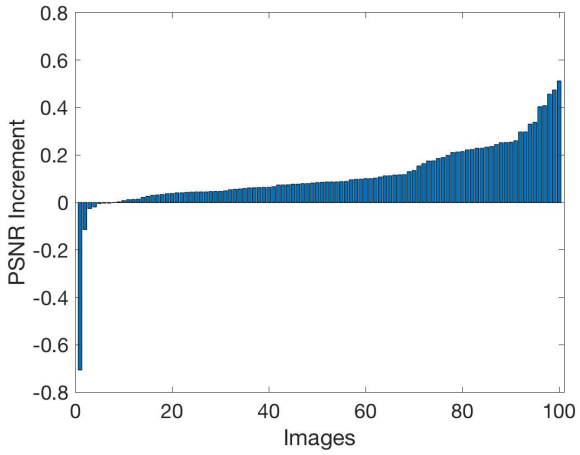

Figure 10: PSNR difference between our improved EDSR and baseline EDSR (sorted by difference in PSNR) on the 100 image validation set from DIV2K (1).

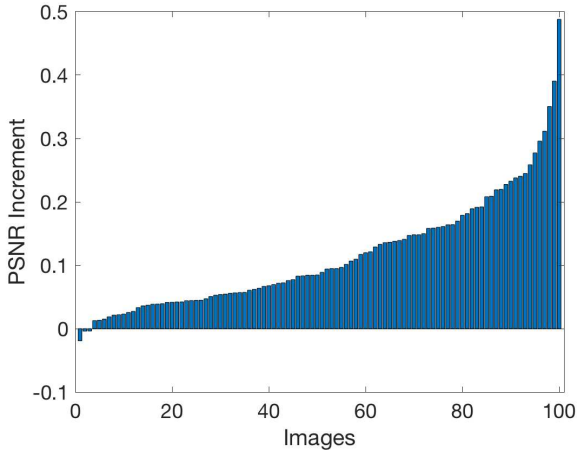

Figure 11: PSNR increment from test time RGB Shuffling (sorted by difference in PSNR)

7. Conclusion

We discussed two new network architectures for denoising and preserving general structure in images during super-resolution, as well as a toolbox of tricks. The vast majority of the findings described here were not implemented in time for the competition deadline. Our high-scoring entries are mostly a result of the toolbox of tricks discussed above. We have noticed substantial improvement from our competition entries to the results reported in this paper.

Our code is available at: `https://github.com/websterbei/EDSR_tensorflow` and `https://github.com/nikhilvravi/DukeSR`. An updated version of this paper will be posted on arXiv.org.

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[2] R. S. Babu and K. S. Murthy. A survey on the methods of super-resolution image reconstruction. *International Journal of Computer Applications*, 15(2), February 2011.

[3] A. Gilman and D. G. Bailey. Near optimal non-uniform interpolation for image super-resolution from multiple images. *Image and Vision Computing New Zealand, Great Barrier Island, New Zealand*, pages 31–35, 2006.

[4] R. C. Hardie, K. J. Barnard, and E. E. Armstrong. Joint map registration and high-resolution image estimation using a sequence of undersampled images. *IEEE transactions on Image Processing*, 6(12):1621–1633, 1997.

[5] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. *Conference on Computer Vision and Pattern Recognition*, 2018.

[6] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, 1993.

[7] S. Kim, N. K. Bose, and H. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(6):1013–1027, 1990.

[8] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *CoRR*, abs/1609.04802, Sept. 2016.

[9] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. *CoRR*, abs/1707.02921, 2017.

[10] A. J. Patti, A. M. Tekalp, and M. I. Sezan. A new motion-compensated reduced-order model kalman filter for space-varying restoration of progressive and interlaced video. *IEEE Transactions on Image Processing*, 7(4):543–554, 1998.

[11] M. Robinson, S. Chiu, J. Lo, C. Toth, J. Izatt, and S. Farsiu. New applications of super-resolution in medical imaging. In *Super-Resolution Imaging*, chapter 1, pages 383–412. CRC Press, 2010.

[12] A. Shocher, N. Cohen, and M. Irani. "Zero-Shot" super-resolution using deep internal learning. *CoRR*, abs/1712.06087, 2017.

[13] R. Timofte, S. Gu, J. Wu, L. Van Gool, L. Zhang, M.-H. Yang, et al. Ntire 2018 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[14] R. Tsai. Multiframe image restoration and registration. *Advance Computer Visual and Image Processing*, 1:317–339, 1984.

[15] H. Ur and D. Gross. Improved resolution from subpixel shifted pictures. *CVGIP: Graphical Models and Image Processing*, 54(2):181–186, 1992.

[16] L. Yue, H. Shen, J. Li, Q. Yuan, H. Zhang, and L. Zhang. Image super-resolution: The techniques, applications, and

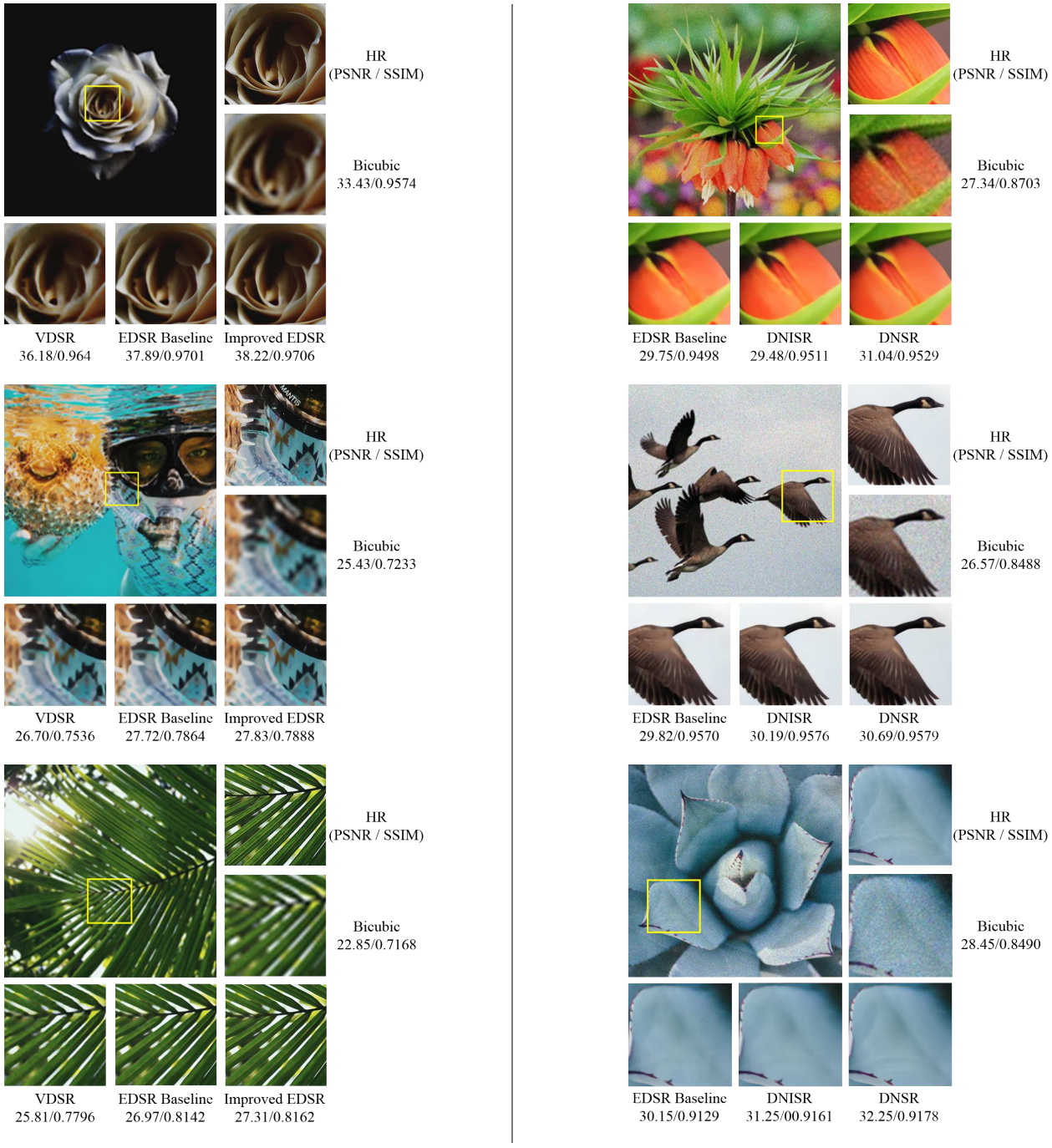

Figure 12: Comparison of different techniques for upscaling images by a factor of 8 (left) and upscaling noisy images by a factor of 4 (right). The visual differences between the images are especially pronounced in the last images of each column, where the folds in the leaves are much clearer.

future. *Signal Processing*, 128:389–408, November 2016.

[17] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *CoRR*, abs/1608.03981, 2016.

[18] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. *CoRR*, abs/1704.03264, 2017.