Supplement: High-Quality Self-Supervised Deep Image Denoising

Samuli Laine NVIDIA
Tero Karras NVIDIA
Jaakko Lehtinen NVIDIA, Aalto University
Timo Aila NVIDIA

A Network architecture, training and evaluation details

Table 1 shows the network architecture used in our blind-spot and baseline networks. This is a slightly modified version of the five-level U-Net [9] architecture that was used by Lehtinen et al. [7]. We add three 1×1 convolution layers at the end in all networks, so that the network depth is the same in both blind-spot and baseline networks. All convolution layers use leaky ReLU [8] with $\alpha = 0.1$, except the very last 1×1 convolution that has linear activation function.

When forming a blind-spot network, we add three additional layers, denoted ROTATE, SHIFT, and UNROTATE in the table. Layer ROTATE forms four rotated versions (by 0°, 90°, 180°, 270°) of the input tensor and stacks them on the minibatch axis. Layer SHIFT pads and shifts every feature map downwards by one pixel, thereby raising the receptive field of every pixel upwards by one pixel. This is needed so that when the receptive fields are later combined, the combination excludes the pixel itself. Finally, layer UNROTATE splits the minibatch axis into four pieces, undoes the rotation done in layer ROTATE, and stacks the results on the channel axis, restoring the minibatch size to the original but quadrupling the feature map count. In addition, in blind-spot networks we modify the convolution layers and downsampling layers to extend their receptive field upwards only, as explained in Section 2 of the paper.

Training and evaluation All networks were initialized following He et al. [3] and trained using Adam with default parameters [5], initial learning rate $\lambda = 0.0003$, and minibatch size of 4. The minibatches were composed of random 256×256 crops from the training set. All networks except those used in impulse noise experiments were trained for 0.5M minibatches, i.e., until 2M training image crops were shown to the network. For the impulse noise experiments we trained the blind-spot networks $2\times$ as long and the baseline networks $8\times$ as long in order to reach convergence. In all training runs, learning rate was ramped down during the last 30% of training using a cosine schedule.

Internally, we use dynamic range of $[0, 1]$ for the image data. The training data was selected to contain only images whose size was between 256×256 and 512×512 pixels, in order to exclude images that were too small for obtaining a training crop, or unnecessarily large compared to the test images. We thus used 44328 training images out of the 50k images in ILSVRC2012 validation set. To run the test images through our rotation-based architecture, each of them was padded to a square shape using mirror padding, denoised, and cropped back to original size. To obtain reliable average PSNRs, we replicated each test set multiple times so that each clean image was corrupted by multiple different instances of noise and, in cases with variable noise parameters, different amounts of noise. Specifically, we replicated test sets KODAK, BSD300, and SET14, by 10, 3, and 20 times, yielding average dataset PSNRs that correspond to averages over 240, 300, and 280 individual denoised images, respectively. All methods were evaluated with the same corrupted input data.

The training runs were executed on NVIDIA DGX-1 servers using four Tesla V100 GPUs in parallel. A typical training run took $\sim$4 hours if using the baseline architecture, and $\sim$14 hours with the blind-spot architecture due to the fourfold increase in minibatch size inside the network. While training we (unnecessarily) computed the mean posterior estimate for every training crop to monitor convergence, performed frequent test set evaluations, etc., which leaves room for optimizing the training speed.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Table 1: Network architecture used in our experiments. Layers marked with $*$ are present only in the blind-spot variants. Layer NIN_A has 384 output feature maps in the blind-spot networks and 96 in the baseline networks.

	NAME	N_{out}	FUNCTION
	INPUT	3	
*	ROTATE	3	Rotate and stack
	ENC_CONV0	48	Convolution 3×3
	ENC_CONV1	48	Convolution 3×3
	POOL1	48	Maxpool 2×2
	ENC_CONV2	48	Convolution 3×3
	POOL2	48	Maxpool 2×2
	ENC_CONV3	48	Convolution 3×3
	POOL3	48	Maxpool 2×2
	ENC_CONV4	48	Convolution 3×3
	POOL4	48	Maxpool 2×2
	ENC_CONV5	48	Convolution 3×3
	POOL5	48	Maxpool 2×2
	ENC_CONV6	48	Convolution 3×3
	UPSAMPLE5	48	Upsample 2×2
	CONCAT5	96	Concatenate output of POOL4
	DEC_CONV5A	96	Convolution 3×3
	DEC_CONV5B	96	Convolution 3×3
	UPSAMPLE4	96	Upsample 2×2
	CONCAT4	144	Concatenate output of POOL3
	DEC_CONV4A	96	Convolution 3×3
	DEC_CONV4B	96	Convolution 3×3
	UPSAMPLE3	96	Upsample 2×2
	CONCAT3	144	Concatenate output of POOL2
	DEC_CONV3A	96	Convolution 3×3
	DEC_CONV3B	96	Convolution 3×3
	UPSAMPLE2	96	Upsample 2×2
	CONCAT2	144	Concatenate output of POOL1
	DEC_CONV2A	96	Convolution 3×3
	DEC_CONV2B	96	Convolution 3×3
	UPSAMPLE1	96	Upsample 2×2
	CONCAT1	99	Concatenate INPUT
	DEC_CONV1A	96	Convolution 3×3
	DEC_CONV1B	96	Convolution 3×3
*	SHIFT	96	Shift down by one pixel
*	UNROTATE	384	Unstack, rotate, combine
	NIN_A	384/96	Convolution 1×1
	NIN_B	96	Convolution 1×1
	NIN_C	9	Convolution 1×1, linear act.

Masking-based training In our training runs with masking-based training (end of Section 4.1), we examine convergence by maintaining a smoothed network whose weights follow the trained network using an exponential moving average. This is a commonly used technique in semi-supervised learning (e.g., [10, 1]) and in evaluating Generative Adversarial Networks (e.g., [2, 4]), and removes the need for a learning rate rampdown — and thus deciding the training length in advance — to measure the results near a local minimum.

All curves in Figure 3 were generated by evaluating the test set using this exponentially smoothed network. We verified in separate tests that the results obtained this way were in line with the usual fixed-length training runs with learning rate rampdown.

B Additional result images

Figures 1, 2 and 3 show additional denoising results for Gaussian, Poisson, and impulse noise, respectively. In these examples the noise model parameters were fixed but unknown for all algorithms. All PSNRs refer to individual images. We recommend zooming in to the images on a computer screen to better view the differences. The full images are also included as PNG files in the supplementary material.

In this larger set of images we can discern some characteristic failure modes of our ablated setups. When the signal covariance $\mathbf{\Sigma}_x$ is forced to be diagonal ("Our ablated, diag. $\mathbf{\Sigma}$"), we can see color artifacts on, e.g., rows 6 and 9 of Figure 1. The diagonal covariance matrix corresponds to having a univariate, independent distribution for each color channel, and therefore the network cannot express being, e.g., certain of hue but uncertain of luminance. This may let the color of noise leak through to the result, as seen in some of the images. With full $\mathbf{\Sigma}_x$ no such color leaking occurs. The ablation which discards information in center pixel entirely ("Our ablated, $\boldsymbol{\mu}$ only") produces strong pixel-scale diamond/checkerboard artifacts, some of which can also be seen in the results of Krull et al. [6]. In images produced by our full, non-ablated method ("Our"), some slight checkerboarding may be seen in high-frequency areas, especially with impulse noise (see, e.g., Figure 3, bottom row). However, in most cases our results are visually indistinguishable from the baseline results.

References

[1] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[2] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[4] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[6] A. Krull, T.-O. Buchholz, and F. Jug. Noise2Void – Learning denoising from single noisy images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2129–2137, 2019.

[7] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2Noise: Learning image restoration without clean data. In *Proc. International Conference on Machine Learning (ICML)*, 2018.

[8] A. L. Maas, A. Y. Hannun, and A. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. International Conference on Machine Learning (ICML)*, 2013.

[9] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 9351:234–241, 2015.

[10] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proc. Advances in Neural Information Processing Systems 30 (NIPS)*, pages 1195–1204. 2017.

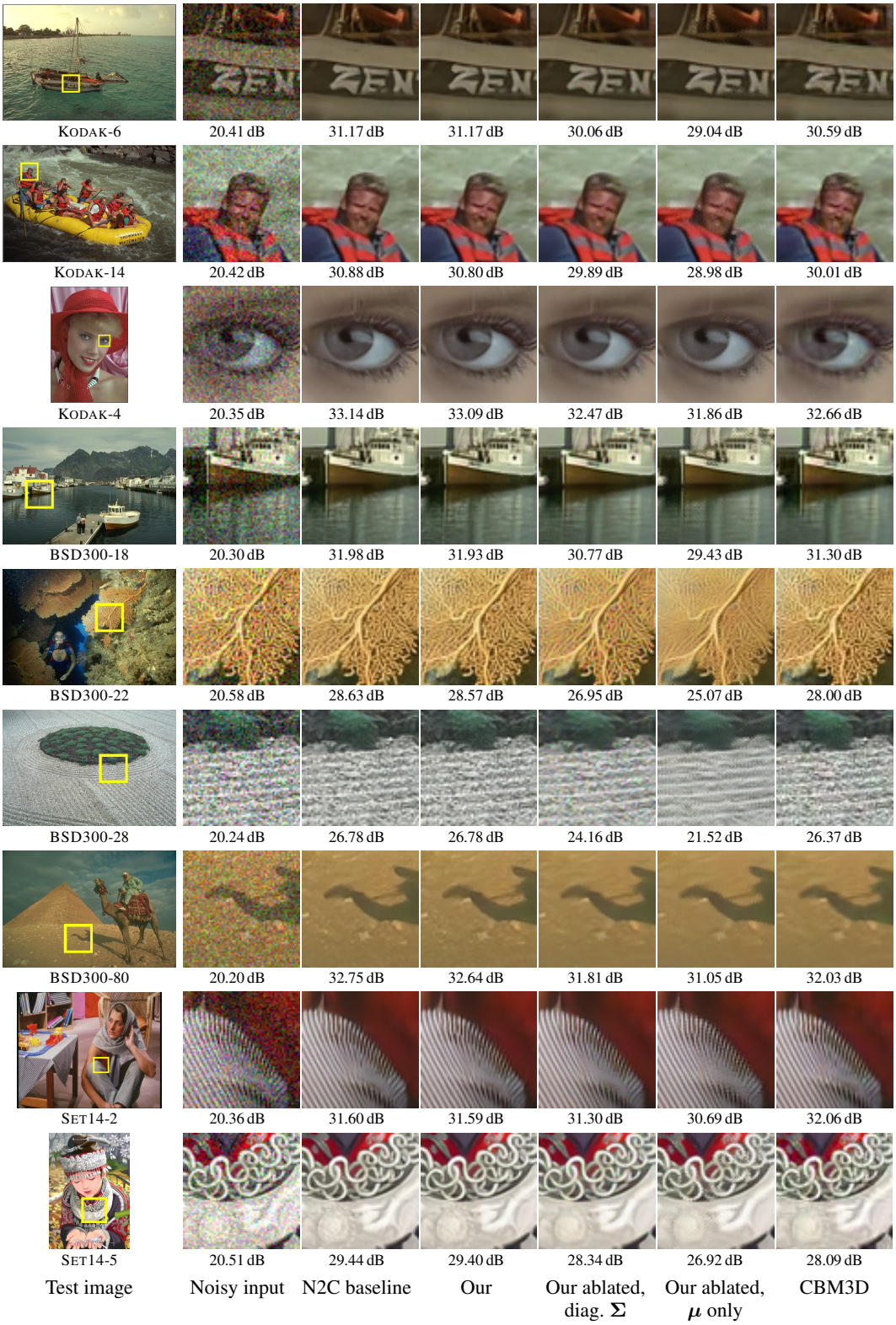

Figure 1: Additional result images for Gaussian noise, $\sigma = 25$.

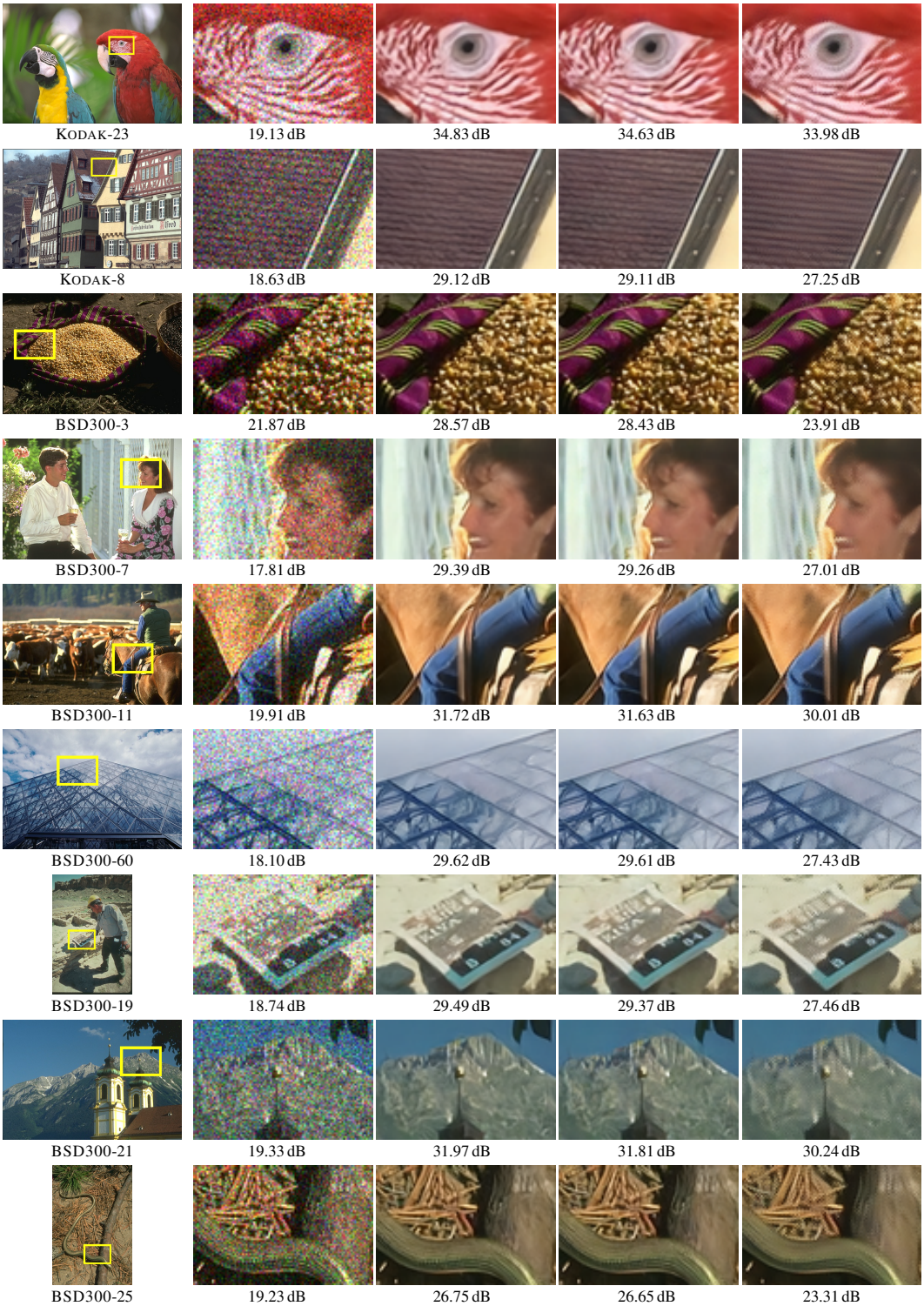

Figure 2: Additional result images for Poisson noise, $\lambda = 30$.

Figure 3: Additional result images for impulse noise, $\alpha = 0.5$.