

# Supplementary Material for “Efficient Learning-based Image Enhancement: Application to Super-resolution and Compression Artifact Removal”

Younghee Kwon<sup>1</sup>

Kwang In Kim<sup>2</sup>

<http://www.mpi-inf.mpg.de/~kkim>

Jin Hyung Kim<sup>3</sup>

<http://ai.kaist.ac.kr/~jkim/>

Christian Theobalt<sup>2</sup>

<http://www.mpi-inf.mpg.de/~theobalt>

<sup>1</sup> Google Inc.

1600 Amphitheatre Parkway,  
Mountain View, CA, USA

<sup>2</sup> Max-Planck-Institut für Informatik

Campus E1-4,  
Saarbrücken, Germany

<sup>3</sup> KAIST

291 Daehak-ro Yuseong-gu,  
Daejeon, Korea

## A Appendix

This appendix presents an intuitive discussion supporting the proposed semi-local GP, especially in the second approximation step, i.e., to use nearest neighbors of a given test input for prediction, discusses details of parameter selection (Sec. A.2), and presents additional experimental results (Sec. A.3).

### A.1 Locality of GP prediction (see Sec. 2 of the main paper)

The idea of using NNs for prediction is based on the assumption that the prediction at  $\mathbf{x}_*$  can be well expressed based on its *local* context. This is partially supported by the fact that for a given test input  $\mathbf{x}_*$ , the predictive distribution of a full GP is represented based on kernel evaluations of  $\mathcal{X}$ :

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}_*)]_j &= \sum_{i=1}^l \alpha_{ij} k(\mathbf{x}_*, \mathbf{x}_i) \\ \mathbb{V}[f(\mathbf{x}_*)] &= k(\mathbf{x}_*, \mathbf{x}_*) - \sum_{i=1}^l \beta_i k(\mathbf{x}_*, \mathbf{x}_i),\end{aligned}\tag{1}$$

where  $\alpha_{ij} = [(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I})^{-1} \mathbf{Y}]_{i,j}$  and  $\beta_i = [(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{f},*}]_i$  and noting that the values of a Gaussian kernel centered at  $\mathbf{x}_*$ ,  $k(\mathbf{x}_*, \cdot)$  diminishes with distance from  $\mathbf{x}_*$ . Accordingly, data points which are distinct from  $\mathbf{x}_*$  (i.e., those lying outside the *effective support* of  $k(\mathbf{x}_*, \cdot)$ ) are unlikely to influence the prediction significantly. However, this argument is incomplete since

the expansion coefficients  $\{\alpha_{ij}, \beta_{ij}\}$  are not taken into account, which is rather complicated due to the matrix inversion term  $(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I})^{-1}$ . Furthermore, the above explanation is applicable to only spatially localized kernels such as Gaussian kernels.

An alternative is to note that by introducing the *equivalent kernel*  $\kappa$  corresponding to  $k$

$$\kappa(\|\mathbf{x}_* - \mathbf{x}_i\|) \triangleq [\mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I})^{-1}]_i, \quad (2)$$

one can restate (1)

$$\begin{aligned} [\mathbb{E}[f(\mathbf{x}_*)]]_j &= \sum_{i=1,\dots,l} \kappa(\|\mathbf{x}_* - \mathbf{x}_i\|) [\mathbf{Y}]_{i,j}, \\ \mathbb{V}[f(\mathbf{x}_*)] &= k(\mathbf{x}_*, \mathbf{x}_*) - \sum_{i=1,\dots,l} \kappa(\|\mathbf{x}_* - \mathbf{x}_i\|) [\mathbf{K}_{\mathbf{f},*}]_i, \end{aligned} \quad (3)$$

which implies that the predictive distribution is specified by two *kernel smoothers*. An interesting property of the equivalent kernel is that it is *spatially localized* (i.e.,  $\kappa(\|\mathbf{x}_* - \cdot\|)$  diminishes quickly with distance from  $\mathbf{x}_*$ ) regardless of the shape of the corresponding kernel  $k$  [10].

In the context of spline smoothing, Silverman [11] showed that there is an asymptotically exact approximation  $\tilde{m}_j(\cdot)$  of  $[\mathbb{E}[f(\mathbf{x}_*)]]_j$ ,

$$[\mathbb{E}[f(\mathbf{x}_*)]]_j \approx \tilde{m}_j(\mathbf{x}_*) = \sum_{i=1}^l \tilde{\kappa}(\|\mathbf{x}_* - \mathbf{x}_i\|) [\mathbf{Y}]_{i,j}, \quad (4)$$

where the corresponding approximate equivalent kernel  $\tilde{\kappa}$  has the localization property. For the case of a Gaussian kernel  $k$ , an analytical approximation has been suggested by Sollich and Williams [12]:

$$\tilde{\kappa}(\|\mathbf{x} - \mathbf{y}\|) = \left( \frac{s_c}{\|\mathbf{x} - \mathbf{y}\|} \right)^{M^2/2} J_{M^2/2}(2\pi s_c \|\mathbf{x} - \mathbf{y}\|), \quad (5)$$

where  $s_c^2 = \log \left( \frac{l(\pi b)^{M^2/2}}{\sigma^2} \right) / (\pi^2 b)$  and  $J$  is the Bessel function of the first kind. This result implies that as  $l$  increases,  $\kappa$  approaches to  $\tilde{\kappa}$ . Furthermore, the support of  $\tilde{\kappa}(\|\mathbf{x}_* - \cdot\|)$  shrinks down and eventually it converges to a single point  $\mathbf{x}_*$  as  $l \rightarrow \infty$ . Accordingly, the localization behavior of  $\kappa$  should be especially prominent when  $l$  is large, which is the case for the current application of image enhancement. Figure 1 shows that even for a relatively small  $l (= 20,000)$ , the qualitative behavior of  $\kappa$  is already in accordance with its analytical approximation  $\tilde{\kappa}$  (i.e.,  $\kappa$  oscillates locally and decays globally) and indeed,  $\kappa$  is strongly localized.

When the output variables  $\mathbf{Y}$  correspond to the pixel-values of images, the variances of its elements are bounded by a constant, which with the locality of  $\kappa$  shows that the *weight functions*  $\{\kappa(\|\mathbf{x}_i - \cdot\|)\}$  corresponding to data points  $\mathbf{x}_i$  which are distinct from  $\mathbf{x}_*$  do not contribute significantly to the expansion (3) and equivalently to (1). For large  $l$ , the expansions become mainly influenced by a subset of  $\mathcal{Y}$ , whose elements have the corresponding inputs which are sufficiently close to  $\mathbf{x}_*$  (say, those inputs in the distance range of  $\varepsilon$  from  $\mathbf{x}_*$ ). In this case, the predictive distribution  $p(f_*|\mathcal{Y}) = p(\mathbf{f}_*|\mathcal{Y}, \mathcal{N}_\varepsilon(f_*))$  should be well-approximated by  $p(\mathbf{f}_*|\mathcal{N}_\varepsilon(f_*))$ , where  $\mathcal{N}_\varepsilon(f_*)$  denotes the values of  $\mathcal{Y}$  for the inputs in the  $\varepsilon$ -neighborhood of  $\mathbf{x}_*$ . Since we do not know the proper values of  $\varepsilon$  in advance, in practice, we simply choose

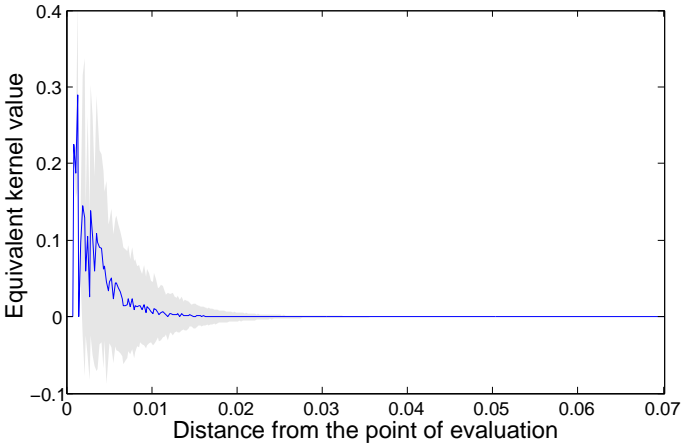


Figure 1: Plot of  $\kappa(r)$  as a function of distance  $r(\cdot) = \|\mathbf{x}_* - \cdot\|^2$  for the data points used in super-resolution experiments: 20,000 training data points are used in calculating  $\mathbf{K}_{\mathbf{f},\mathbf{f}}$  while a distinct set of 20,000 data points are used as the evaluation points  $\{\mathbf{x}_*\}$ . The gray area corresponds to twice the standard deviations.

$n$ -NNs of  $\mathbf{x}_*$  ( $C_n(\mathbf{x}_*)$ ). Experiments on the quality of semi-local GP approximation can be found in [8].

The localization property of equivalent kernels  $\kappa$  has been demonstrated not only for localized kernels  $k$  (e.g., Gaussian) but also for non-local kernels such as thin-plate spline kernel [8], sigmoid kernels, and polynomial kernels [10]. Accordingly, the proposed scheme can also be applied to non-local kernels.

## A.2 Parameter selection

There are several parameters to be tuned, which include the input and output patch sizes ( $M$ ) and ( $N$ ), the regularization parameter ( $\sigma_P$ ), and the (hyper-)parameters for regression (the noise variance  $\sigma^2$ , the kernel parameter  $b$ , and the numbers of training and inducing inputs  $n$  and  $m(|C_m|)$ , respectively). The parameters  $m$  and  $n$  were fixed at 50 and 200, respectively. These values are determined by trading the performance against the computational complexity. We observed steady performance increases as  $m$  and  $n$  increase. At the same time, the run-time grew roughly quadratically and linearly with respect to  $m$  and  $n$ , respectively. The remaining parameters were optimized based on a set of validation images which are disjoint from the images used in training and testing.

## A.3 Additional Results

This section presents additional experimental results which include example images and quantitative performance comparisons by PSNR. For the enhancement of JPEG images, we present results corresponding to the Q3 compression quality of Table 2 in [10] (cf. main paper) (Figs. 2-5 and 7). We also present a comparison with Laparra et al.’s support vector regression (SVR)-based algorithm [9] for JPEG artifact removal (Figs. 6 and 7). To enable a fair comparison with this algorithm, only six gray level images (from the 7-th to

12-th images in Fig. 1 of the corresponding paper) were used. These images were cropped to  $256 \times 256$ -pixels around the center and were encoded using Matlab’s ‘*imwrite*’ function with compression quality 7 (as was done in [8]).<sup>1</sup> In this set of experiments, the hyper-parameters of the proposed method optimized for Q2 (cf. main paper and Table 2 of [8]) were re-used as suggested by our on-line scenario.

For single-image super-resolution, results for magnification factors 2 and 3 along each dimension are presented in Figs. 8-10. For a magnification factor of 3, we additionally present a comparison with another state-of-the-art learning-based approach proposed by Yang et al. [10].<sup>2</sup>

We suggest viewing the electronic version of this document for a clearer visualization of the results.

### A.3.1 Enhancement of JPEG images

The SVR-based method [8] successfully removed block artifacts and produced sharp edges (cf. the last two columns of Fig. 6). However, it tended to leave ringing artifacts and accordingly, made the results more noisy than other methods. The results of the proposed method are almost as sharp as the results of SVR at the edges. Furthermore, our results show fewer ringing artifacts as our method coherently reconstructs sharp edges and texture details. This is clearly visible in the visor of Lena (Fig. 2) and in the helmet of the biker in Fig. 4. The shape-adaptive DCT (SADCT) demonstrated comparable performance to the proposed method in terms of PSNR. However, detailed visual inspection (e.g. in the visor of Lena in Fig. 2 and at the stripe pattern of the tiger in Fig. 5) reveals that the results of the proposed method are much cleaner (with fewer ringing artifacts) and contain more detail. For quantitative evaluation, improvements in PSNR values of the results of different algorithms with respect to input JPEG images are plotted in Fig. 7.

### A.3.2 Super-resolution

All tested super-resolution algorithms produced significantly sharper images than bicubic resampling. However, the results of Freeman et al.’s algorithm [9] and Yang et al.’s algorithm [10] are noisy as observed in Lena (Fig. 8) and the tiger and the parrot in Fig. 9. This is not the case for the results of the proposed method and Kim and Kwon’s algorithm [4]. Furthermore, the proposed method and [4] produced as sharp images as the results of [9], thereby improving significantly both the visual quality and the PSNR values (Fig. 10).

A major advantage of the proposed method over Kim and Kwon’s algorithm [4] is that the semi-local GP used in the proposed method has two orders of magnitude faster training time over the sparse approximation of [4], which enables the system to be quickly adapted to the distribution of a specific class of images, as exemplified in document image super-resolution (Fig. 2 of the main paper). By training the system only on document images, the proposed method produced much cleaner images than the output of the sparse method [4] trained based on generic images.

<sup>1</sup>This experimental setting is not the fundamental limitation of Laparra et al.’s algorithm. This setting enables us to directly use the code kindly shared by the authors at [http://www.uv.es/vista/vistavalencia/denoising\\_SVR/index.html](http://www.uv.es/vista/vistavalencia/denoising_SVR/index.html).

<sup>2</sup>We used the code and the corresponding dictionary (i.e., a set of basis vectors used in representing the images, trained for a magnification factor of 3, see [10] for details) kindly made available by the authors at <http://www.ifp.illinois.edu/~jyang29/>.



Figure 2: Example of artifact suppression for *Lena* image: a. input JPEG images, b. re-application of JPEG [10], c. SADCT [10], d. the proposed method, and e. original un-encoded images.



Figure 3: Example of artifact suppression for *Goldhill* image: a. input JPEG images, b. re-application of JPEG [10], c. SADCT [10], d. the proposed method, and e. original un-encoded images.



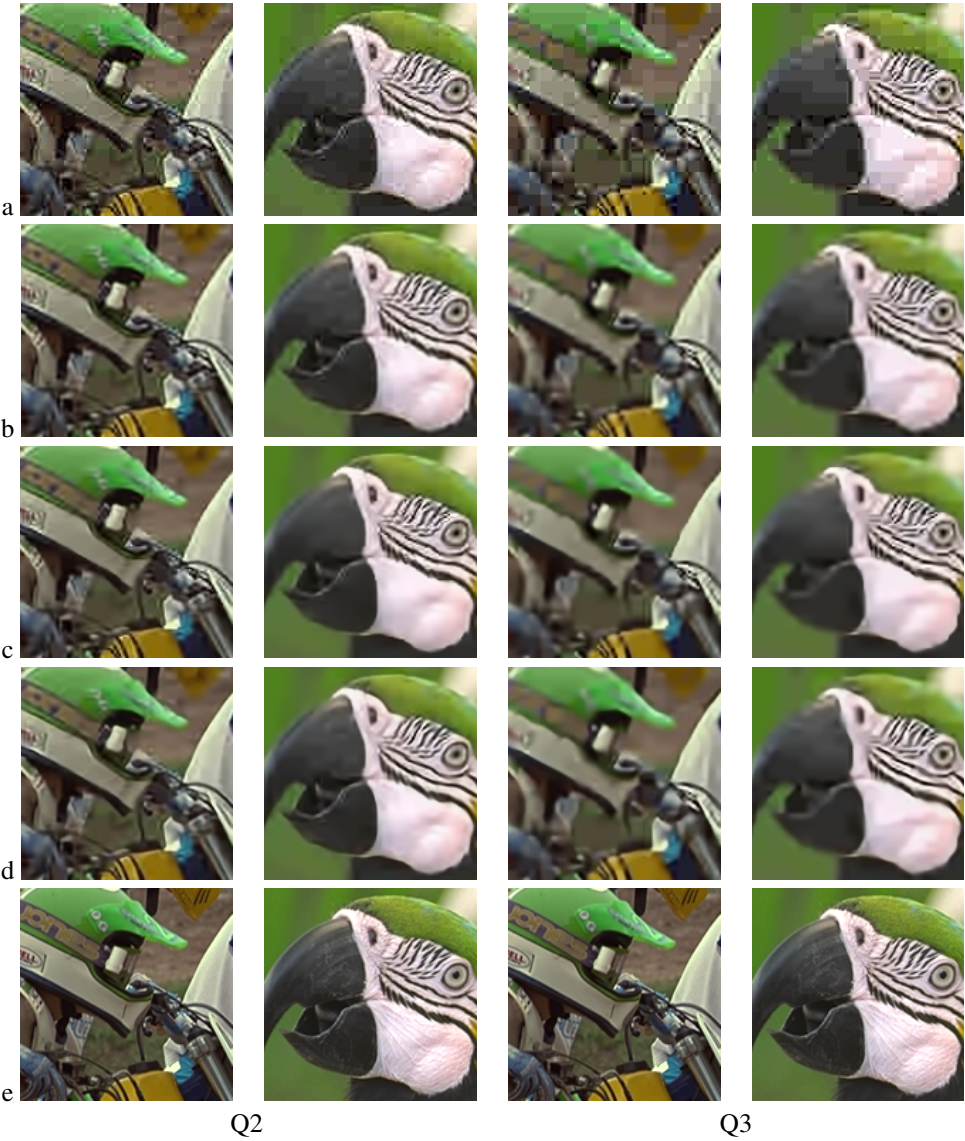


Figure 4: Examples of artifact suppression: a. input JPEG images, b. re-application of JPEG [1], c. SADCT [1], d. the proposed method, and e. original un-encoded images.

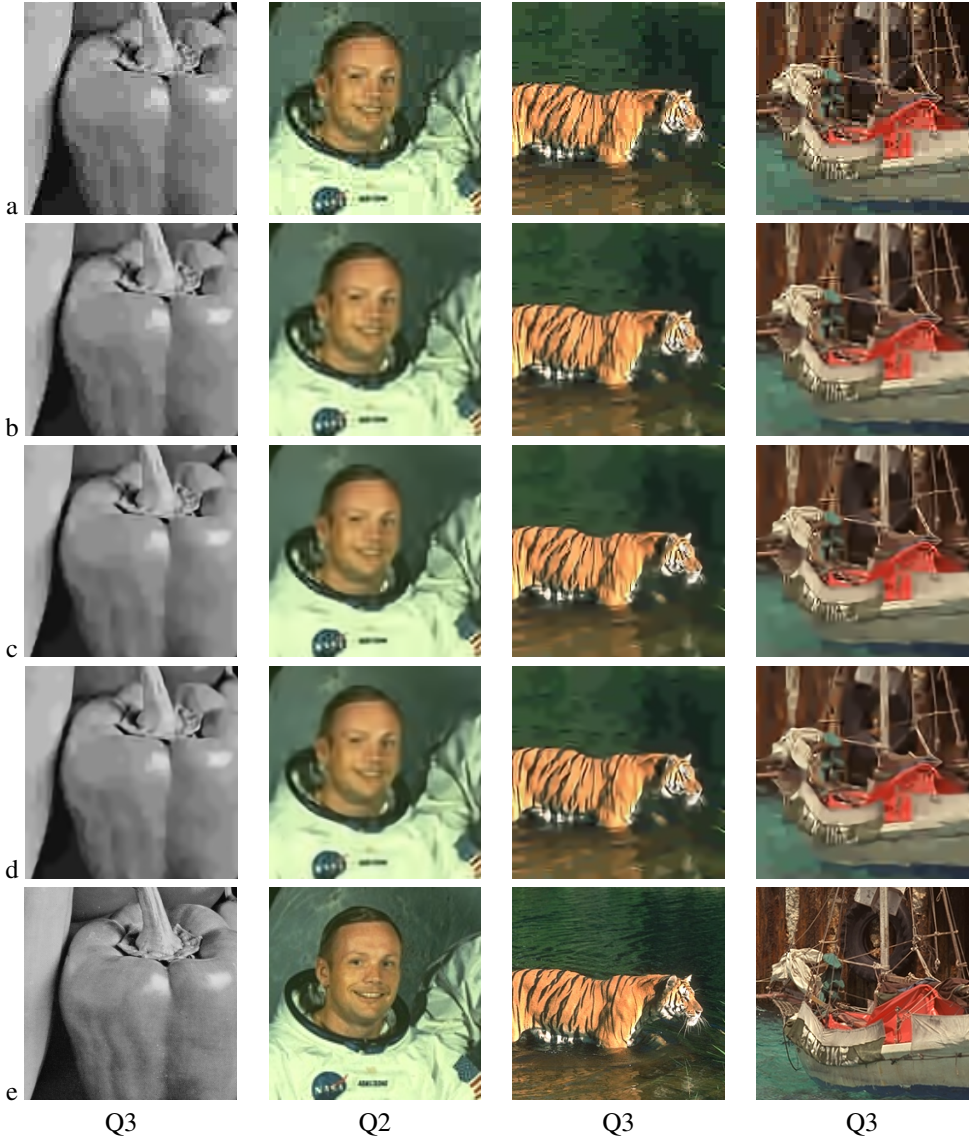


Figure 5: Examples of artifact suppression: a. input JPEG images, b. re-application of JPEG [10], c. SADCT [10], d. the proposed method, and e. original un-encoded images.





Figure 6: Comparison of the proposed method on  $(256 \times 256)$ -size sub-images: a. input JPEG images, b. SVR-based method [4], c. SADCT [4], d. the proposed method, and e. original un-encoded images.

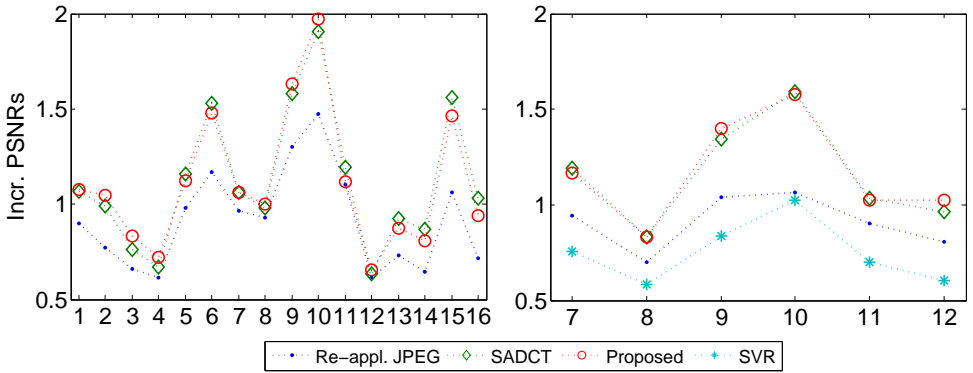


Figure 7: Performance of different JPEG enhancement algorithms. Left: results for JPEG images compressed using the quantization table Q3. Right: results for  $(256 \times 256)$ -size subimages compressed with quality 7. The  $x$  axis corresponds to the image index (see Fig. 1 of the main paper).

## References

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [2] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Trans. Image Processing*, 16(5):1395–1411, 2007.
- [3] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.
- [4] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010.
- [5] K. I. Kim, Y. Kwon, J.-H. Kim, and C. Theobalt. Efficient learning-based image enhancement: application to compression artifact removal and super-resolution. Technical Report MPI-I-2011-4-002, Max-Planck-Institut für Informatik, February 2011.
- [6] V. Laparra, J. Gutiérrez, G. Camps-Valls, and J. Malo. Image denoising with kernels based on natural image relations. *Journal of Machine Learning Research*, 11:873–903, 2010.
- [7] A. Nosratinia. Denoising of JPEG images by re-application of JPEG. *Journal of VLSI Signal Processing*, 27(1):69–79, 2001.
- [8] B. W. Silverman. Spline smoothing: the equivalent variable kernel method. *The Annals of Statistics*, 12(3):898–916, 1984.
- [9] P. Sollich and C. K. I. Williams. Using the equivalent kernel to understand Gaussian process regression. In *Advances in Neural Information Processing Systems*, 2005.

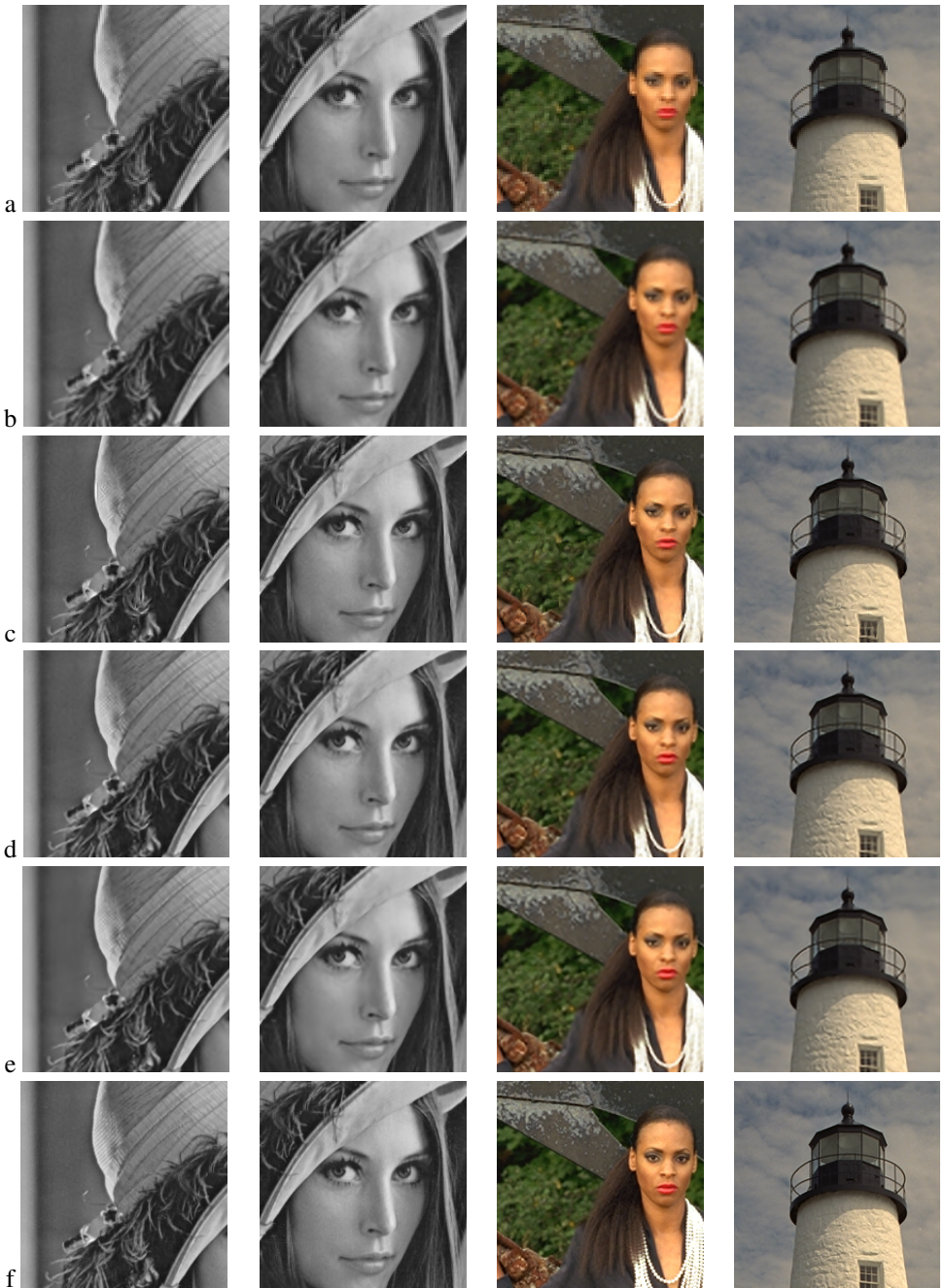


Figure 8: Examples of image super-resolution (magnification factor 2): a. low-resolution image displayed in the target scale (nearest neighbor resampling), b. bicubic resampling, c. Freeman et al. [8], d. Kim and Kwon [9], e. the proposed method, and f. ground truth high-resolution images.



Figure 9: Examples of image super-resolution (magnification factor 3): a. nearest neighbor resampling, b. bicubic resampling, c. Yang et al. [14], d. Kim and Kwon [8], e. the proposed method, and f. ground truth high-resolution images.



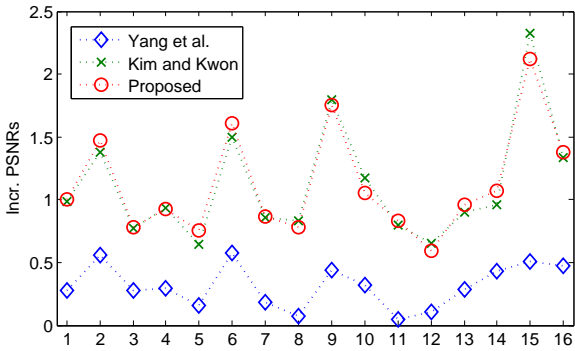


Figure 10: Performance of different image super-resolution algorithms (magnification factor 3).

[10] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Processing*, 19(11):2861–2873, 2010.