Plug-and-Play Image Deblurring and Super-Resolution with Vision Transformer Denoiser

1 Introduction

The purpose of image restoration is to recover the latent clean image $\mathbf{x}$ from its degraded observation $\mathbf{y} = \mathcal{T}(\mathbf{x}) + \mathbf{n}$, where $\mathcal{T}$ is the noise-irrelevant degradation operation, $\mathbf{n}$ is assumed to be additive white Gaussian noise (AWGN) of standard deviation σ. By specifying different degradation operations, Different image restoration tasks can be achieved. Typical image restoration tasks include image denoising when $\mathcal{T}$ is an identity operation or image deblurring when $\mathcal{T}$ is a two-dimensional convolution operation.

Since image restoration is an inverse problem, the solution $\hat{\mathbf{x}}$ can be obtained by solving a Maximum A Posteriori (MAP) estimation problem,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}), \tag{1}$$

where $\log p(\mathbf{y}|\mathbf{x})$ represents the log-likelihood of observation $\mathbf{y}$, $\log p(\mathbf{x})$ models the prior of clean image $\mathbf{x}$. (1) can be reformulated as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{y} - \mathcal{T}(\mathbf{x})\|^2 + \lambda\mathcal{R}(\mathbf{x}), \tag{2}$$

where the solution minimizes to a data term $\frac{1}{2\sigma^2}\|\mathbf{y} - \mathcal{T}(\mathbf{x})\|^2$ and a regularization term $\lambda\mathcal{R}(\mathbf{x})$ with regularization parameter λ. The data term guarantees the solution is consistent with the degradation process, while the prior term enforces desired property on the solution.

The methods to solve (2) can be divided into two main categories, i.e., model-based methods and learning-based methods. Model-based methods aim to directly solve (2) with some optimization algorithms, which are flexible to handle various IR tasks by simply specifying $\mathcal{T}$ and can directly optimize on the degraded image y but they are usually time-consuming. Learning-based methods train a model through an optimization of a loss function on a training set containing degraded-clean image pairs, this process requires cumbersome training to learn the model before testing and are usually restricted to specialized tasks. However, they have a fast testing speed but also tend to deliver better performance due to the end-to-end training.

The two categories of methods have their respective pros and cons, Plug and Play (PnP) methods can be used to leverage their respective pros by replacing the denoising

subproblem of model-based optimization with learning-based denoisers such as CNNs or Vision Transformers with the help of variable splitting algorithms such as alternating direction method of multipliers (ADMM) or halfquadratic splitting (HQS) or iterative shrinkage thresholding algorithm (ISTA). This work emplyos these three variabel splitting algorithms on Image deblurring and single-image superresolution.

2 Half Quadratic Splitting (HQS) Algorithm

In order to decouple the data term and prior term of (2), HQS introduces an auxiliary variable $\mathbf{z}$, resulting in a constrained optimization problem given by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \ \frac{1}{2\sigma^2}\|\mathbf{y} - \mathcal{T}(\mathbf{x})\|^2 + \lambda\mathcal{R}(\mathbf{z}) \quad s.t. \quad \mathbf{z} = \mathbf{x}. \tag{3}$$

(3) is then solved by minimizing the following problem

$$\mathcal{L}_\mu(\mathbf{x}, \mathbf{z}) = \frac{1}{2\sigma^2}\|\mathbf{y} - \mathcal{T}(\mathbf{x})\|^2 + \lambda\mathcal{R}(\mathbf{z}) + \frac{\mu}{2}\|\mathbf{z} - \mathbf{x}\|^2, \tag{4}$$

where μ is a penalty parameter. Such problem can be addressed by iteratively solving the following subproblems for $\mathbf{x}$ and $\mathbf{z}$ while keeping the rest of the variables fixed,

$$\mathbf{x}_k = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathcal{T}(\mathbf{x})\|^2 + \mu\sigma^2\|\mathbf{x} - \mathbf{z}_{k-1}\|^2 \tag{5}$$

$$\mathbf{z}_k = \arg\min_{\mathbf{z}} \frac{1}{2(\sqrt{\lambda/\mu})^2}\|\mathbf{z} - \mathbf{x}_k\|^2 + \mathcal{R}(\mathbf{z}). \tag{6}$$

The data term and prior term are decoupled into two separate subproblems. The subproblem of (5) usually has a fast closed-form solution depending on $\mathcal{T}$, while the subproblem of (6) corresponds to Gaussian denoising on $\mathbf{x}_k$ with noise level $\sqrt{\lambda/\mu}$. Consequently, any Gaussian denoiser can be plugged into the alternating iterations to solve (2). Equation (6) can be rewritten as follows

$$\mathbf{z}_k = Denoiser(\mathbf{x}_k, \sqrt{\lambda/\mu}). \tag{7}$$

A individual CNN or Vision transformer can be learned to replace (7) to exploit the advantages of deep networks.

3 Alternating Direction Method of Multipliers (ADMM)

Equation (2) can be written in the form of a typical MAP optimization problem where f(x) is the negative log likelihood and g(x) is the negative log prior probability as

$$\hat{x} = \arg\min_x \{f(x) + g(x)\} \tag{8}$$

Similar to HQS, we split the variable into two variables, x and z, with the constraint that x = v. So then the new, but fully equivalent, problem is given by

$$\hat{x} = \arg\min_x \{f(x) + g(z)\} \quad s.t. \quad \mathbf{z} = \mathbf{x}. \tag{9}$$

In order to enforce the constraint, An additional term is added to the cost function that penalizes large differences between x and v. The expression being minimized in equation (10) is known as the augmented Lagrangian for the constrained optimization problem and the term u serves a role equivalent to that of a Lagrange multiplier.

$$(\hat{x}, \hat{z}) = \arg\min_{(x,z)} \left\{ f(x) + g(z) + \frac{a}{2}\|x - z + u\|^2 \right\} \tag{10}$$

The above equation can now be simplified to the optimization of (x, z) in two steps using alternating minimization of x and z in an iterative fashion as shown below

$$\hat{x} \leftarrow \arg\min_{x} \left\{ f(x) + \frac{a}{2}\|x - \hat{z} + u\|^2 \right\} \tag{11}$$

$$\hat{z} \leftarrow \arg\min_{z} \left\{ g(z) + \frac{a}{2}\|\hat{x} - z + u\|^2 \right\} \tag{12}$$

$$u \leftarrow u + (\hat{x} - \hat{z}) \tag{13}$$

Equation (12) can be replaced by a CNN or a Vision Transformer based denosier as shown below

$$\mathbf{z}_k = Denoiser(\mathbf{x}_k). \tag{14}$$

4 Iterative Soft-Thresholding Algorithm (ISTA)

Considering the following optimization problem represented by equation 8. The update step using proximal gradient method is given by equation 15.

$$x_{k+1} := \mathrm{Prox}_{\eta g}\left(x_k - \eta \nabla f\left(x_k\right)\right) \tag{15}$$

Where the proximal operator is defined as follows

$$\mathrm{Prox}_{\lambda g}(\boldsymbol{x}) = \underset{\boldsymbol{v}}{\operatorname{argmin}} \frac{1}{2\lambda}\|\boldsymbol{v} - \boldsymbol{x}\|^2 + g(\boldsymbol{v}) \tag{16}$$

Replacing the proximal operator by a denoiser we get the final update step as

$$x_{k+1} := Denoiser\left(x_k - \eta \nabla f\left(x_k\right)\right) \tag{17}$$

5 Vision Transformers

Vision Transformer models have consistently obtained state-of-the-art results in computer vision tasks, including object detection and video classification. In contrast to standard CNNs, which cannot model long range pixel dependencies due to its limited receptive field, Vision Transformers model global attention. However, the primary disadvantage of vision transformers is image scalability. The computational complexity of self attention in Vision transformers grows quadratically with the spatial resolution, therby prohibiting its application to high-resolution images.

To overcome this computational bottle neck, A recent vision transformer architecture called Restomer [6] has been implemented as an image denoiser in this work. It applies self attention across feature dimension rather than the spatial dimension which leads to linear complexity. The Restomer model also achieves the current state-of-the-art results on image denoising.

The Restomer artiecture has been trained synthetic benchmark datasets generated with additive white Gaussian noise (Set12 [3], BSD68 [2] , Urban100 [1], Kodak24 and McMaster [5]) as well as on real-world datasets (SIDD [6] and DND [4]). The models were trained on three noise levels 15, 25 and 50 and A blind gaussian denoiser with noise randomly sampled between [0,50] has also been trained.

6 Image Deblurring

The degradation model for deblurring a blurry image with uniform blur (or image deconvolution) is generally expressed as

$$\mathbf{y} = \mathbf{x} \otimes \mathbf{k} + \mathbf{n} \tag{18}$$

where $\mathbf{x} \otimes \mathbf{k}$ denotes two-dimensional convolution between the latent clean image $\mathbf{x}$ and the blur kernel $\mathbf{k}$. By assuming the convolution is carried out with circular boundary conditions, the fast solution of (5) is given by

$$\mathbf{x}_k = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{y}) + \alpha_k \mathcal{F}(\mathbf{z}_{k-1})}{\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{k}) + \alpha_k}\right) \tag{19}$$

where the $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote Fast Fourier Transform (FFT) and inverse FFT, $\overline{\mathcal{F}(\cdot)}$ denotes complex conjugate of $\mathcal{F}(\cdot)$.

7 Single Image Super-Resolution (SISR)

The degradation model is assumed to be a low-resolution (LR) image, Which is a blurred, decimated, and noisy version of high-resolution (HR) image. The mathematical formulation is given by

$$\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}, \tag{20}$$

where $\downarrow_s$ denotes a s-fold downsampler, i.e., selecting the upper-left pixel for each distinct $s \times s$ patch.

By assuming the convolution is carried out with circular boundary conditions as in deblurring, the closed-form solution is given by

$$\mathbf{x}_k = \mathcal{F}^{-1}\left(\frac{1}{\alpha_k}\Big(\mathbf{d} - \overline{\mathcal{F}(\mathbf{k})} \odot_s \frac{(\mathcal{F}(\mathbf{k})\mathbf{d}) \Downarrow_s}{(\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{k})) \Downarrow_s + \alpha_k}\Big)\right), \tag{21}$$

where $\mathbf{d} = \overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{y} \uparrow_s) + \alpha_k \mathcal{F}(\mathbf{z}_{k-1})$ and where $\odot_s$ denotes applying element-wise multiplication to the $s \times s$ distinct blocks of $\overline{\mathcal{F}(\mathbf{k})}$, $\Downarrow_s$ denotes distinct block downsampler, by averaging the $s \times s$ distinct blocks

Table 1: Average PSNR(dB) image deblurring performance of Restomer and DRUnet on 8 kernels

Method	$K1$	$K2$	$K3$	$K4$	$K5$	$K6$	$K7$	$K8$
DRUnet	30.41	29.82	29.91	29.73	31.31	31.13	30.26	29.76
Restomer	29.28	29.00	29.08	28.69	30.38	29.91	27.67	28.99

8 Experiments on Image Deblurring

8.1 Image Deblurring using HQS

The blurry images are generated by first applying a blur kernel and then adding additive Gaussian noise with noise level σ. Eight blur kernels are applied and Gaussian noise with a standard deviation of 7.65 is added to the image after blurring.

A Restomer trained with various noise levels in $[0, 50]$ is plugged in place of (6). λ is set to 0.075. The number of iterations (k) is set to 8. σ_k is uniformly sampled from 49 to 330 in log space. μ_k is determined by the relation $\mu_k = \lambda/\sigma_k^2$. The results of the Restomer architecture and the current state of the art in PnP image deblurring, DRUnet [4] has been repored on table 1 and the results of deblurring of the first kernel has been represented on figure 1.

8.2 Image Deblurring using ADMM

The blurry images are generated by first applying a Gaussian blur kernel and then adding additive Gaussian white noise with a standard deviation of 7.65 . Three denoisers trained with 15, 25, 50 noise level respectively and a blind denoiser trained on randomly sampled noise between [0,50] has been used on three images from the set12 dataset. The results have been reported on Table 2, 3, 4. The figures of the output have been plotted on figure 4.

It can be observed from the Butterfly figures that the estimated images produced by the denoiser trained on noise level 50 is much smoother compared to other images. The number of iterations reacquired tends to decrease as the noise level of the denoiser increases but the performance also decreases. It can also be observed that the blind denoiser recquires the maximum number of iterations to stabilize.

8.3 Image Deblurring using ISTA

Similar procedure as deblurring with ADMM has been applied. The results have been reported on Table 5, 6, 7. The figures of the output have been plotted on figure 4.

Compared to image deblurring using ADMM, ISTA requires more iterations to converge on image deblurring. The amount of time required per iterative step is similar on both the algorithms. ADMM takes 0.17ms per iteration whereas ISTA takes 0.18ms

Table 2: Average PSNR(dB) image deblurring performance of Restomer Denoisers on Butterfly image using ADMM

Denoiser	Rho	iterations	PSNR
15	0.22	21	24.27
25	0.26	12	24.58
50	0.1	9	23.67
Blind	1	22	23.51

Table 3: Average PSNR(dB) image deblurring performance of Restomer Denoiser on Starfish image using ADMM

Denoiser	Rho	iterations	PSNR
15	0.22	8	26.26
25	0.21	5	26.1
50	0.1	9	24.85
Blind	1	15	25.72

Table 4: Average PSNR(dB) image deblurring performance of Restomer Denoiser on Leaves image using ADMM

Denoiser	Rho	iterations	PSNR
15	0.28	25	24.56
25	0.2	17	24.52
50	0.07	9	24.62
Blind	1	31	22.12

Table 5: Average PSNR(dB) image deblurring performance of Restomer Denoisers on Butterfly image using ISTA

Denoiser	Rho	iterations	PSNR
15	1.8	76	24.50
25	1.4	33	24.95
50	1.8	14	23.76
Blind	1.8	13	23.63

Table 6: Average PSNR(dB) image deblurring performance of Restomer Dnoisers on Starfish image using ISTA

Denoiser	Rho	iterations	PSNR
15	1.8	18	26.24
25	1.8	19	26.41
50	1.8	18	25.90
Blind	1.6	31	25.82

per iteration. Unlike in the case of ADMM, blind denoiser on ISTA takes the minimum number of iterations and the number of iterations required tends to decrease as the noise level of the denoiser increases . Both the algorithms achieve similar performance on the images and achieve their highest performance on the starfish image. The performance at each iterative step for noise level 50 denoiser on the starfish image using ADMM and ISTA has been represented at figure 3. It can be seen that the performance of ISTA improves at each iterative step, unlike ADMM.

9 Experiments on Image Super-Resolution

9.1 Super-Resolution using ISTA

The low resolution images were generated by applying a Gaussian blur kernel followed by downsampling the image by a factor of two. Additive Gaussian white noise with a standard deviation of 7.65 was then added to the image. Three denoisers trained with 15, 25, 50 noise level respectively and a blind denoiser trained on randomly sampled noise between [0,50] has been used on three images from the set12 dataset. The results

Table 7: Average PSNR(dB) image deblurring performance of Restomer Dnoisers on Leaves image using ISTA

Denoiser	Rho	iterations	PSNR
15	1.9	39	24.76
25	1.9	32	24.52
50	1.8	22	24.48
Blind	1.8	17	23.31

Table 8: Average PSNR(dB) Super-Resolution performance of Restomer Denoisers on Butterfly image using ISTA

Denoiser	Rho	iterations	PSNR
15	2.3	46	26.70
25	2	24	26.41
50	7.4	23	26.64
Blind	3	6	25.19

Table 9: Average PSNR(dB) Super-Resolution performance of Restomer Denoisers on Starfish image using ISTA

Denoiser	Rho	iterations	PSNR
15	2.9	17	28.11
25	3.3	7	27.89
50	7.5	21	28.26
Blind	5.8	1	26.94

have been reported on Table 8, 9, 10. The figures of the output have been plotted on figure 5.

It can be observed that, contrary to the case of image deblurring using ISTA, the number of iterations required to achieve optimal performance decreases as the noise level of the denoiser increases while using ISTA. The number of iterations required by the blind denoiser is significantly lower compared to other denoisers coupled with ISTA in Super-resolution

9.2 Super-Resolution using ADMM

Similar procedure as Super-Resolution with ISTA has been applied. The results of Butterfly has been reported on Table 11. The figures of the output has been plotted on figure 6.

Contrary to the case of image deblurring, The number of iterations required to achieve optimal performance on Super-resolution is similar between ADMM and ISTA.

Table 10: Average PSNR(dB) Super-Resolution performance of Restomer Denoisers on Leaves image using ISTA

Denoiser	Rho	iterations	PSNR
15	2.5	33	27.35
25	2.6	28	26.46
50	7.9	18	27.21
Blind	6	3	24.81

Table 11: Average PSNR(dB) Super-Resolution performance of Restomer Denoisers on Butterfly image using ADMM

Denoiser	Rho	iterations	PSNR
15	0.4	45	26.51
25	0.4	20	25.8
50	0.1	18	26.77
Blind	0.6	12	24.43

Table 12: Average PSNR(dB) Super-Resolution performance of Blind Restomer Denoiseron images using HQS

Image	λ	iterations	PSNR
Butterfly	10	24	27.67
Leaves	10	24	27.51
StarFish	10	24	30.72

Unlike the case of image deblurring using ADMM, The number of iterations reacquired to achieve optimal performance decreases as the noise level of the denoiser increases.The blind denoiser requires the lowest number of iterations similar to Super resolution using ADMM.

9.3 Super-Resolution using HQS

Similar procedure as Super-Resolutio with ISTA has been applied.

A Restomer trained with various noise levels in $[0, 50]$ is plugged in place of (6). λ is set to 10. The number of iterations (k) is set to 24. σ_k is uniformly sampled from 49 to 0 in log space. μ_k is determined by the relation $\mu_k = \lambda/\sigma_k^2$. The results of the Restomer architecture has been repored on table 12 and the results of Super-Resolutio has been represented on figure 7.

References

[1] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. "Single Image Super-resolution from Transformed Self-Exemplars". In: June 2015. DOI: 10.1109/CVPR.2015.7299156.

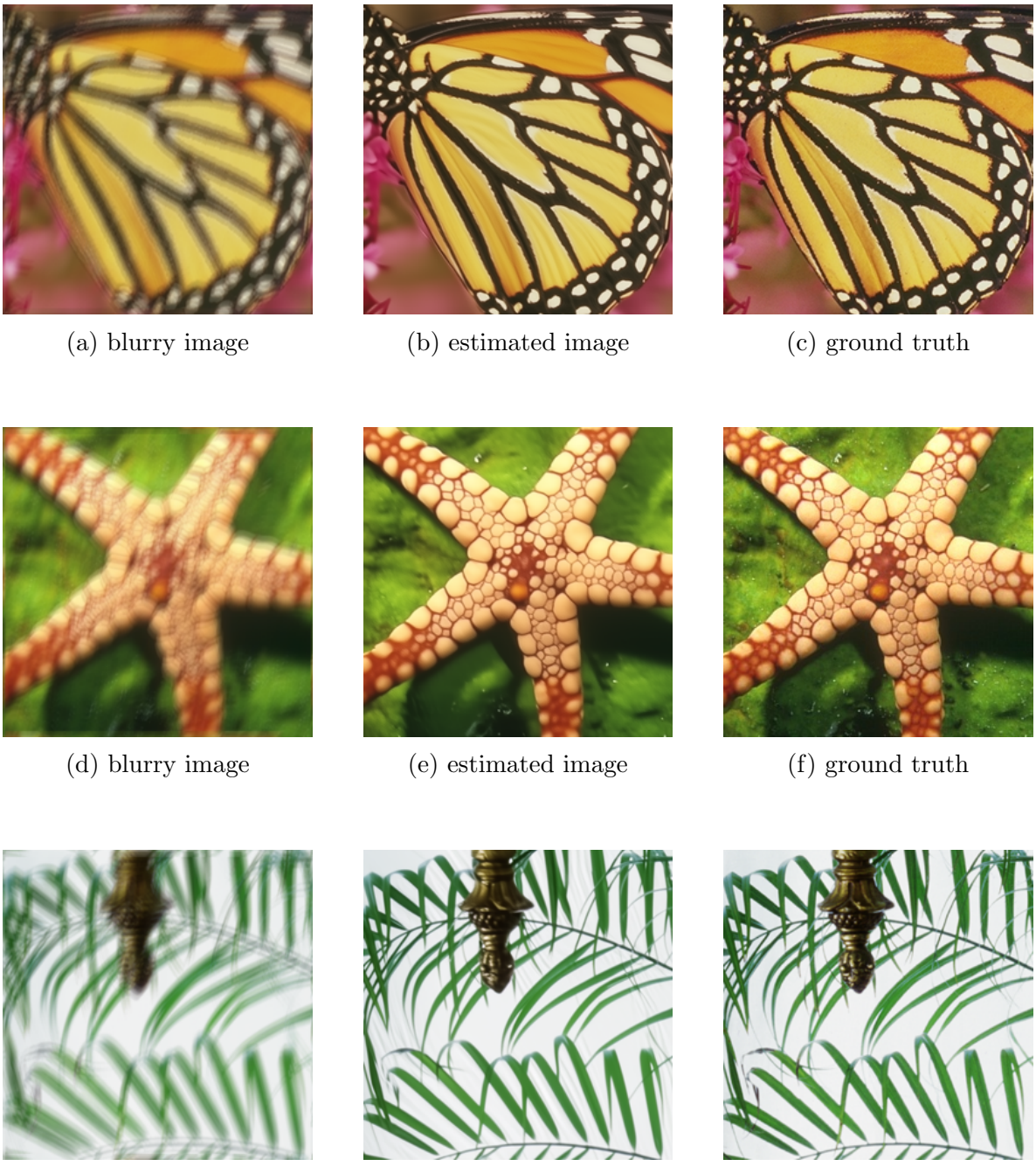
[2] David Martin et al. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics". In: vol. 2. Feb. 2001, 416–423 vol.2. ISBN: 0-7695-1143-0. DOI: 10.1109/ICCV.2001.937655.

[3] Kai Zhang et al. "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising". In: *IEEE Transactions on Image Processing* 26.7 (July 2017), pp. 3142–3155. DOI: 10.1109/tip.2017.2662206. URL: https://doi.org/10.1109%2Ftip.2017.2662206.

[4] Kai Zhang et al. "Plug-and-Play Image Restoration with Deep Denoiser Prior". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[5] Lei Zhang et al. *Color Demosaicking by Local Directional Interpolation and Non-local Adaptive Thresholding.*

[6] Feida Zhu et al. "Blind Face Restoration via Integrating Face Shape and Generative Priors". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* June 2022, pp. 7662–7671.



(a) blurry image (b) estimated image (c) ground truth

(d) blurry image (e) estimated image (f) ground truth

(g) blurry image (h) estimated image (i) ground truth

Figure 1: Visual results of image deblurring by Restomer using HQS

(a) ground truth - Butterfly

(b) ground truth - Star

(c) ground truth - Leaves

(d) Estimated image by noise level 15 denoiser

(e) Estimated image by noise level 15 denoiser

(f) Estimated image by noise level 15 denoiser

(g) Estimated image by noise level 25 denoiser

(h) Estimated image by noise level 25 denoiser

(i) Estimated image by noise level 25 denoiser

(j) Estimated image by noise level 50 denoiser

(k) Estimated image by noise level 50 denoiser

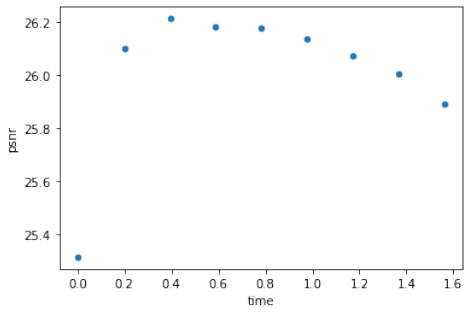
(l) Estimated image by noise level 50 denoiser

(m) Estimated image by blind denoiser

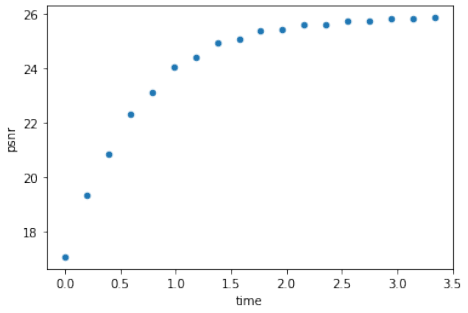
(n) Estimated image by blind denoiser

(o) Estimated image by blind denoiser

Figure 2: Visual results of image deblurring by Restomer using ADMM



(a) noise level 50 denoiser on starfish image using ADMM



(b) noise level 50 denoiser on starfish image using ISTA

Figure 3: iterative performance of ADMM and ISTA on startfish image

(a) ground truth - Butterfly

(b) ground truth - Star

(c) ground truth - Leaves

(d) Estimated image by noise level 15 denoiser

(e) Estimated image by noise level 15 denoiser

(f) Estimated image by noise level 15 denoiser

(g) Estimated image by noise level 25 denoiser

(h) Estimated image by noise level 25 denoiser

(i) Estimated image by noise level 25 denoiser

(j) Estimated image by noise level 50 denoiser

(k) Estimated image by noise level 50 denoiser

(l) Estimated image by noise level 50 denoiser

(m) Estimated image by blind denoiser

(n) Estimated image by blind denoiser

(o) Estimated image by blind denoiser

Figure 4: Visual results of image deblurring by Restomer using ISTA

(a) ground truth - Butterfly

(b) ground truth - Star

(c) ground truth - Leaves

(d) Estimated image by noise level 15 denoiser

(e) Estimated image by noise level 15 denoiser

(f) Estimated image by noise level 15 denoiser

(g) Estimated image by noise level 25 denoiser

(h) Estimated image by noise level 25 denoiser

(i) Estimated image by noise level 25 denoiser

(j) Estimated image by noise level 50 denoiser

(k) Estimated image by noise level 50 denoiser

(l) Estimated image by noise level 50 denoiser

(m) Estimated image by blind denoiser

(n) Estimated image by blind denoiser

(o) Estimated image by blind denoiser

Figure 5: Visual results of image Super-Resolution by Restomer using ISTA

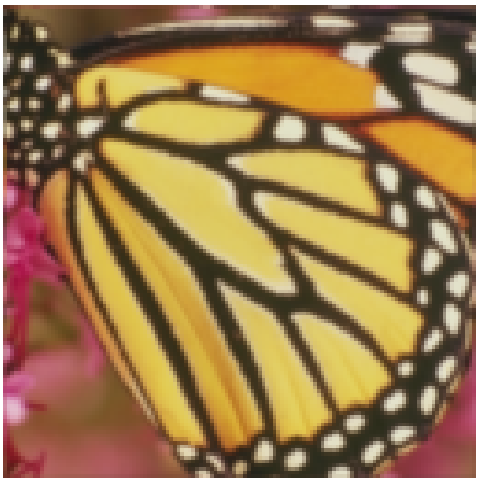
(a) Estimated image by noise level 15 denoiser

(b) Estimated image by noise level 25 denoiser

(c) Estimated image by noise level 50 denoiser

(d) Estimated image by blind denoiser

Figure 6: Visual results of image Super-Resolution by Restomer using ADMM



(a) Low resolution image of Butterfly

(b) Estimated image of Butterfly

(c) Low resolution of Leaves

(d) Estimated image of Leaves

(e) Low resolution of Starfish

(f) Estimated image of Starfish

Figure 7: Visual results of image Super-Resolution by Restomer using HQS