Plug-and-Play Image Restoration with Deep Denoiser Prior

Kai Zhang Yawei Li Wangmeng Zuo Lei Zhang Luc Van Gool
Radu Timofte

Presenter:
Yoel Bokobza

Outline

1. Motivation & Background
2. Derivation of the Method
    a. Half Quadratic Splitting (HQS)
    b. Iterative Solution
    c. The Denoiser - DRUNet
    d. Relationship to the Course Material
3. Results

Motivation & Background

01

Introduction

- Image restoration is the operation of taking a corrupted image y and estimating the latent clean image x
- Corruption forms: noise, blurring, down sampling, mosaicing
- The corrupted image model: $y = \mathcal{T}(x) + n$
- n – Additive white Gaussian noise (AWGN) of variance σ^2
- $\mathcal{T}(\cdot)$ – Noise irrelevant degradation operation

Objective Function

- The solution $\hat{x}$ can be obtained by solving a Maximum A Posteriori (MAP) estimation problem $\hat{x} = \arg\max_x \log p(x|\mathrm{y}) = \arg\max_x \log p(y|x) + \log p(x)$

- Objective Reformulation:

$$\hat{x} = \arg\min_x \frac{1}{2\sigma^2} ||y - \mathcal{T}(x)||^2 + \lambda \Re(x)$$

- $\frac{1}{2\sigma^2} ||y - \mathcal{T}(x)||^2$ - Data term

- $\Re(x)$ – regularization term - enforces desired properties of the output

- λ - regularization parameter

Derivation of the Method (DPIR)

02

Half Quadratic Splitting (HQS) Method

- Decoupling the data term and the regularization term
- Adding an auxiliary variable z, resulting in a constrained optimization problem:

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

- HQS method tries to minimize the following cost function:

$$L_\mu(x, z) = \frac{1}{2\sigma^2} \|y - \mathcal{T}(x)\|^2 + \lambda \mathfrak{R}(z) + \frac{\mu}{2} \|z - x\|^2$$

- μ – penalty parameter

Iterative Solution

$\arg\min_{x,z} \frac{1}{2\sigma^2} \|y - \mathcal{T}(x)\|^2 + \lambda\Re(z) + \frac{\mu}{2}\|z - x\|^2$ can be addressed by alternating optimization:

$$x_k = \arg\min_{x} \|y - \mathcal{T}(x)\|^2 + \mu\sigma^2\|x - z_{k-1}\|^2$$

$$z_k = \arg\min_{z} \frac{1}{2\left(\sqrt{\lambda/\mu}\right)^2}\|z - x_k\|^2 + \Re(z)$$

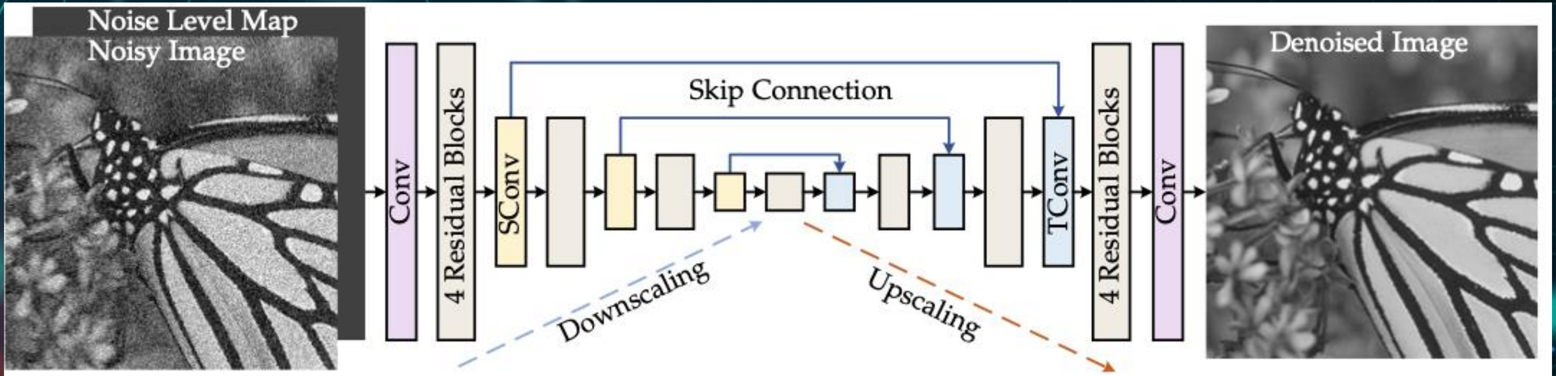
- The first subproblem usually has a fast closed-form solution depending on $\mathcal{T}$
- The second subproblem corresponds to Gaussian denoising image x_k with noise level $\sqrt{\lambda/\mu}$

Methodology for Parameter Setting

- To guarantee x_k and z_k converge to a fixed point, a large μ is needed
- The strategy: gradually increase μ, resulting in a sequence of $\mu_1 < \cdots < \mu_K$
- μ_k controls the noise level $\sigma_k = \sqrt{\frac{\lambda}{\mu_k}}$
- Actually, the $\{\sigma_k\}_{k=1}^{K}$ are decreased from σ_1 to σ_K in log space

Denoising Network DRUNet

- $z_k = \arg\min_z \frac{1}{2(\sqrt{\lambda/\mu_k})^2} ||z - x_k||^2 + \Re(z) \triangleq Denoiser(x_k, \sqrt{\lambda/\mu_k})$
- Solving the Denoiser problem is hard due to the lack of the prior $\Re(z)$
- A good solution is to implement a neural network
- DRUNet takes an additional noise level map as input and combines U-Net and ResNet



Relationship to the Course Material

Plug-and-Play ADMM (Lecture 8)

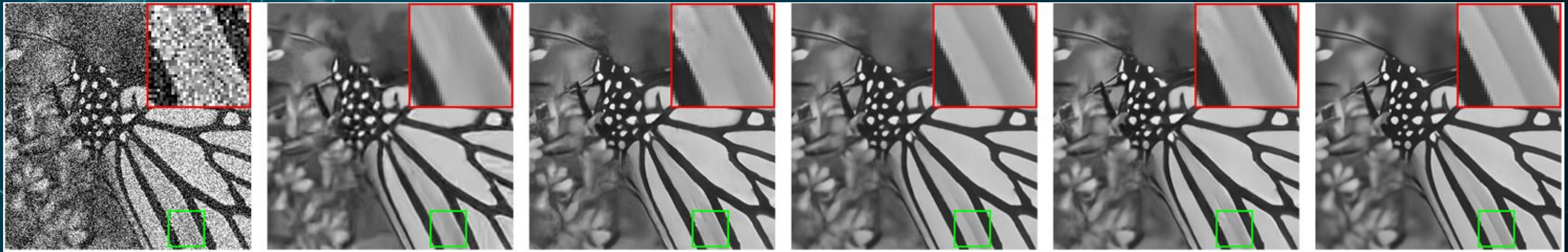
- Both ADMM (Lecture 2) and HQS are used for variable splitting
- ADMM is a method that seeks the saddle point through the primal-dual formulation
- In both methods, the proximal mapping can be interpreted as a Denoiser

03

Results

Denoising Results

Grayscale Image Denoising



(a) Noisy (14.78dB) (b) BM3D (25.82dB) (c) DnCNN (26.83dB) (d) RNAN (27.18dB) (e) FFDNet (26.92dB) (f) DRUNet (27.31dB)

Color Image Denoising

(a) Noisy (14.99dB) (b) BM3D (28.36dB) (c) DnCNN (28.68dB) (d) FFDNet (28.75dB) (e) IRCNN (28.69dB) (f) DRUNet (29.28dB)

Image Deblurring

- The blurry image is expressed by:

$$y = x \otimes k + n$$

- $\mathcal{T}(x) = x \otimes k$ - 2D convolution between the latent clean image x and the blur kernel k

- A closed form solution for the first iterative optimization problem exists

Image Deblurring - Results

(a) Blurry image (b) DMPHN (9.74dB) (c) FDN (23.53dB) (d) IRCNN (28.45dB) (e) IRCNN+ (28.14dB) (f) DPIR (30.27dB)

Fig. 7. Visual results comparison of different deblurring methods on *Leaves*. The blur kernel is visualized in the upper right corner of the blurry image. The noise level is 7.65(3%).

(a) $\mathbf{x}_1$ (16.34dB) (b) $\mathbf{z}_1$ (23.75dB) (c) $\mathbf{x}_4$ (22.33dB) (d) $\mathbf{z}_4$ (29.37dB) (e) $\mathbf{x}_8$ (29.34dB) (f) Convergence curves

Fig. 8. (a)-(e) Visual results and PSNR results of $\mathbf{x}_k$ and $\mathbf{z}_k$ at different iterations; (f) Convergence curves of PSNR (y-axis) for $\mathbf{x}_k$ and $\mathbf{z}_k$ with respect to number of iterations (x-axis).

Single Image Super-Resolution (SISR)

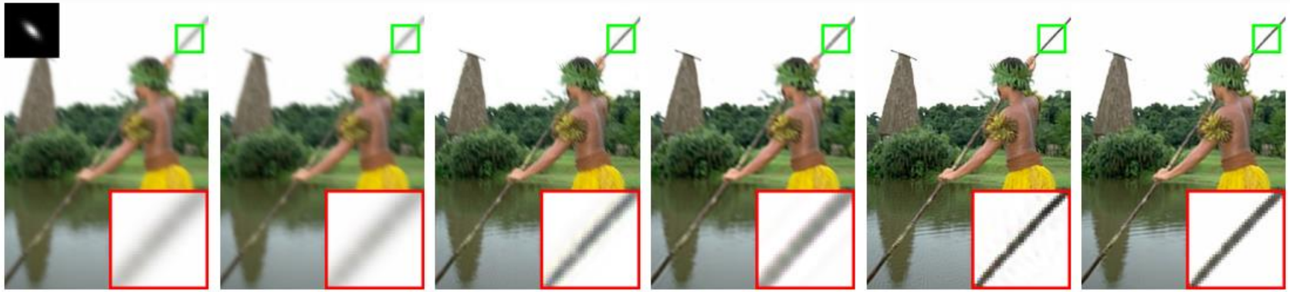
- The degradation model is expressed by:

$$y = (x \otimes k) \downarrow_s + n$$

- $\downarrow_s (\cdot)$ - standard s-fold downsampler, selecting the upper-left pixel for each distinct $s \times s$ patch

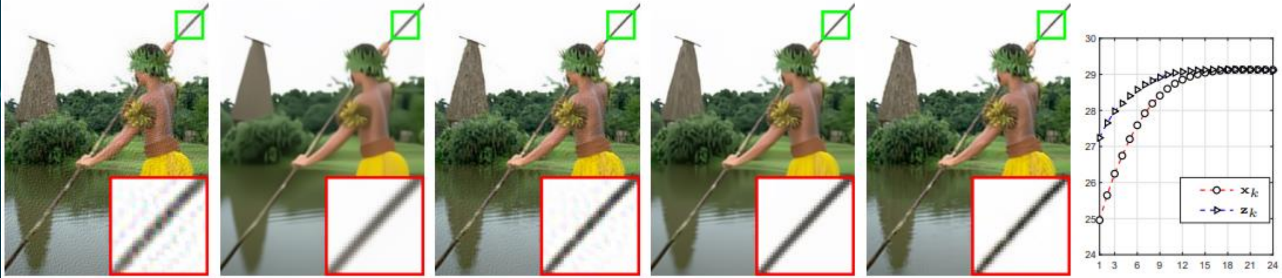
- A closed form solution for the first iterative optimization problem exists

SISR- Results



(a) Bicubic (24.82dB) (b) RCAN (24.61dB) (c) MZSR (27.34dB) (d) IRCNN (26.89dB) (e) IRCNN+ (28.65dB) (f) DPIR (29.12dB)

Fig. 11. Visual results comparison of different SISR methods on an image corrupted by classical degradation model. The kernel is shown on the upper-left corner of the bicubicly interpolated LR image. The scale factor is 2.



(a) $\mathbf{x}_1$ (24.95dB) (b) $\mathbf{z}_1$ (27.24dB) (c) $\mathbf{x}_6$ (27.59dB) (d) $\mathbf{z}_6$ (28.57dB) (e) $\mathbf{x}_{24}$ (29.12dB) (f) Convergence curves

Fig. 12. (a)-(e) Visual results and PSNR results of $\mathbf{x}_k$ and $\mathbf{z}_k$ at different iteration; (f) Convergence curves of PSNR results (y-axis) for $\mathbf{x}_k$ and $\mathbf{z}_k$ with respect to number of iterations (x-axis).

Image Demosaicing

- The degradation model is expressed by:
$$y = M \odot x$$
- $\odot$ - denotes element wise multiplication
- M is a matrix with binary elements indicating the missing pixels of y
- A closed form solution for the first iterative optimization problem exists

Image Demosaicing - Results

(a) Ground-truth (b) Matlab (33.67dB) (c) DDR (41.94dB) (d) DeepJoint (42.49dB) (e) MMNet (40.62dB) (f) RNAN (43.77dB)

(g) LSSC (42.31dB) (h) IRI (39.49dB) (i) FlexISP (36.95dB) (j) IRCNN (40.18dB) (k) IRCNN+ (40.85dB) (l) DPIR (43.23dB)

Fig. 13. Visual results comparison of different demosaicing methods on image *kodim19* from Kodak dataset.