Medical Image Analysis and Processing

Image Noise Filtering

Sparse Denoising

Emad Fatemizadeh

Distance/online Course: Session 16

Date: 20 April 2021, 31th Farvardin 1400

Contents

Low-Rank Matric Recovery

› Most popular approach is low rank minimization problem

$$\widehat{\boldsymbol{X}} = \min_{X} \left\{ \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 + \lambda \|\boldsymbol{X}\|_* \right\}$$

› Where $\|\boldsymbol{X}\|_*$ is nuclear norm:

$$\|\boldsymbol{X}\|_* = \sum_{i=1}^{r} \sigma_i (\boldsymbol{X})$$

› It is shown that the basic solution is:

› $\boldsymbol{X} = \boldsymbol{U\Sigma_\lambda V^T}$, where $\mathbf{Y} = \boldsymbol{U\Sigma V^T}$(SVD of noisy observation) and:

$$\boldsymbol{\Sigma}_\lambda = \max\{\boldsymbol{\Sigma} - \lambda \boldsymbol{I}, \mathbf{0}\}$$

Low-Rank Matric Recovery

› **W**eighted **N**uclear **N**orm **M**inimization (WNNM) is an alternative:

$$\widehat{\boldsymbol{X}} = \min_{X}\left\{\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}\|_F^2 + \|\boldsymbol{X}\|_{w_*}\right\}$$

› where $\|\boldsymbol{X}\|_{\boldsymbol{w}_*}$ is weighted nuclear norm:

$$\|\boldsymbol{X}\|_{\boldsymbol{w}_*} = \sum_{i=1}^{r} w_i\sigma_i\,(\boldsymbol{X}), \qquad 0 \le w_1 \le w_2 \le \cdots \le w_r$$

› It is shown that the solution is:

› $\boldsymbol{X} = \boldsymbol{U\Sigma_\lambda V^T}$, where $\mathbf{Y} = \boldsymbol{U\Sigma V^T}$(SVD of noisy observation) and:

$$\boldsymbol{\Sigma_\lambda} = \max\{\boldsymbol{\Sigma} - diag(\boldsymbol{w}), \boldsymbol{0}\}$$

WNNM Denoising

- An important fact:
- The singular values of a matrix are always sorted in a non-ascending order, and the larger singular values usually correspond to the subspaces of more important components of the data matrix.
- Therefore, we would better shrink the larger singular values *less*, that is, assigning smaller weights to the larger singular values in the weighted nuclear norm.

WNNM Denoising Idea

- Algorithm:
- For a local patch $\boldsymbol{y}_j$ in image $\boldsymbol{y}$, we search for its nonlocal similar patches across the image.
- Stacking those nonlocal similar patches into a matrix, denote by $\boldsymbol{Y}_j$, we have $\boldsymbol{Y}_j = \boldsymbol{X}_j + \boldsymbol{N}_j$, where $\boldsymbol{X}_j$ and $\boldsymbol{N}_j$ are the patch matrices of original image and noise, respectively.
- Intuitively, $\boldsymbol{X}_j$ is a low rank matrix, and the low rank matrix approximation methods can be used to estimate $\boldsymbol{X}_j$ from $\boldsymbol{Y}_j$.
- By aggregating all the denoised patches, the whole image can be estimated.

WNNM Denoising Formulation

- Using the noise variance σ_n^2 to normalize the F-norm:

$$\widehat{\boldsymbol{X}_j} = \min_{\boldsymbol{X}_j} \left\{ \frac{1}{\sigma_n^2} \left\| \boldsymbol{X}_j - \boldsymbol{Y}_j \right\|_F^2 + \left\| \boldsymbol{X}_j \right\|_{w_*} \right\}$$

- Key problem is weight vector determination!
- In the application of denoising, the larger the singular values, the less they should be shrunk.

MEDICAL IMAGE ANALYSIS AND PROCESSING, E.FATEMIZADEH (2020)

WNNM Denoising Formulation

› Therefore, a natural idea is:

$$w_i = \frac{c\sqrt{n}}{\sigma_i(\boldsymbol{X}_j) + \varepsilon}$$

› where $c > 0$ is a constant, n is the number of similar patches in $\boldsymbol{Y}_j$, $\varepsilon = 10^{-6}$.

› The singular values $\sigma_i(\boldsymbol{X}_j)$ are not available. We assume that the noise energy is evenly distributed over each subspace spanned by the basis pair of **U** and **V**:

$$\hat{\sigma}_i(\boldsymbol{X}_j) = \sqrt{max(\sigma_i^2(\boldsymbol{Y}_j) - n\sigma_n^2, 0)}$$

Low-Rank Matric Recovery

› WNNM Algorithm:

Algorithm 1 Image Denoising by WNNM

Input: Noisy image $\boldsymbol{y}$

Initialize $\hat{\boldsymbol{x}}^{(0)} = \boldsymbol{y}, \boldsymbol{y}^{(0)} = \boldsymbol{y}$

for k=1:K **do**

  Iterative regularization $\boldsymbol{y}^{(k)} = \hat{\boldsymbol{x}}^{(k-1)} + \delta(\boldsymbol{y} - \hat{\boldsymbol{y}}^{(k-1)})$

  for each patch $\boldsymbol{y}_j$ in $\boldsymbol{y}^{(k)}$ **do**

    Find similar patch group $\boldsymbol{Y}_j$

    Estimate weight vector $\boldsymbol{w}$

    Singular value decomposition $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] = SVD(\boldsymbol{Y}_j)$

    Get the estimation: $\hat{\boldsymbol{X}}_j = \boldsymbol{U}\mathcal{S}_{\boldsymbol{w}}(\boldsymbol{\Sigma})\boldsymbol{V}^T$

  end for

  Aggregate $\boldsymbol{X}_j$ to form the clean image $\hat{\boldsymbol{x}}^{(k)}$

end for

Output: Clean image $\hat{\boldsymbol{x}}^{(K)}$

Fidelity Criteria

› How to measure distance between denoised and clean image, or denoising performance.

› Consider additive noise model:

$$\boldsymbol{y}(i,j)=\boldsymbol{x}(i,j)+\boldsymbol{n}(i,j)$$

› and $\widehat{\boldsymbol{x}}(i,j)$ is denoised image, where $\{\boldsymbol{x},\boldsymbol{y},\boldsymbol{n},\widehat{\boldsymbol{x}}\} \in \mathbb{R}^{M\times N}$

› Most used (with reference) metrics are:

- SNR
- PSNR
- SSIM

Fidelity Criteria

› Signal-to-Noise Ratio (in dB)

$$SNR(x,y) = 10log_{10}\frac{\sum_{i,j}\left(\boldsymbol{x}(i,j)\right)^2}{\sum_{i,j}\left(\boldsymbol{x}(i,j)-\widehat{\boldsymbol{x}}(i,j)\right)^2}$$

› Higher means better (maximum is infinite in theory)

› around 40dB is OK

Fidelity Criteria

› Peak Signal-to-Noise Ratio (in dB)

$$PSNR(x,y) = 10log_{10}\frac{L^2}{\frac{1}{MN}\sum_{i,j}\left(\boldsymbol{x}(i,j)-\hat{\boldsymbol{x}}(i,j)\right)^2}$$

› where L is peak signal value in image (for example: 255)

› Higher means better (maximum is infinite in theory)

› Around 40dB is OK

Fidelity Criteria

› The Structural SIMilarity (SSIM) Index)

› This metric correlated to human perception

$$SSIM\left(x(i,j),\hat{x}(i,j)\right)=\frac{\left(2\mu_{x(i,j)}\mu_{\hat{x}(i,j)}+C_1\right)\left(2\sigma_{x(i,j)\hat{x}(i,j)}+C_2\right)}{\left(\mu_{x(i,j)}^2+\mu_{\hat{x}(i,j)}^2+C_1\right)\left(\sigma_{x(i,j)}^2+\sigma_{\hat{x}(i,j)}^2+C_2\right)}$$

› where $\mu_{x(i,j)}$; $\mu_{\hat{x}(i,j)}$; $\sigma_{x(i,j)}^2$, and $\sigma_{\hat{x}(i,j)}^2$ are the means and variances of a window around pixel (i,j), respectively, and $\sigma_{x(i,j)\hat{x}(i,j)}$ is the covariance between windows around (i,j) in x and $\hat{x}$, C_1 and C_2 are constant values used to avoid instability.

Fidelity Criteria

- The Structural SIMilarity (SSIM) Index
- $C_1 = (k_1 L)^2$ and $C_1 = (k_2 L)^2$,
- $k_1 = 0.01, k_2 = 0.03$, as default
- Maximum (best match) is 1.0
- Overall SSIM Index of two images:

$$\mathrm{MSSIM}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{\#\ of\ windows} \sum_{(i,j)} SSIM\big(\boldsymbol{x}(i,j), \hat{\boldsymbol{x}}(i,j)\big)$$

Fidelity Criteria

- Less used fidelity criteria (with references):
  - EPI (Edge Preserving Index)
  - FSIM (Feature similarity index)
  - QILV (Quality Index based on Local Variance)
  - IQI (Image Quality Index)
  - MI (Mutual Information)
- No reference fidelity criteria:
  - SI (Sharpness Index) using image total variation.
  - Image Entropy (More is better)
  - NIQE (Naturalness Image Quality Evaluator)

Machine Learning

- The most recent methods uses Deep Neural Network
- Deep Neural Network (CNN) is a complex and highly nonlinear mapping (deep cascade of convolution layer) from input space to output space: $\mathbb{R}^{M\times N} \to \mathbb{R}^{P\times Q}$
- $\boldsymbol{x}$: input, $\boldsymbol{y}$: output, Θ: map parameters:

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}; \Theta)$$

- Deep Neural Network (CNN) training, is to estimate desired output with low error with respect to a loss function:

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\, loss(\mathcal{F}(\boldsymbol{x}; \Theta), \boldsymbol{y}_{desired})$$

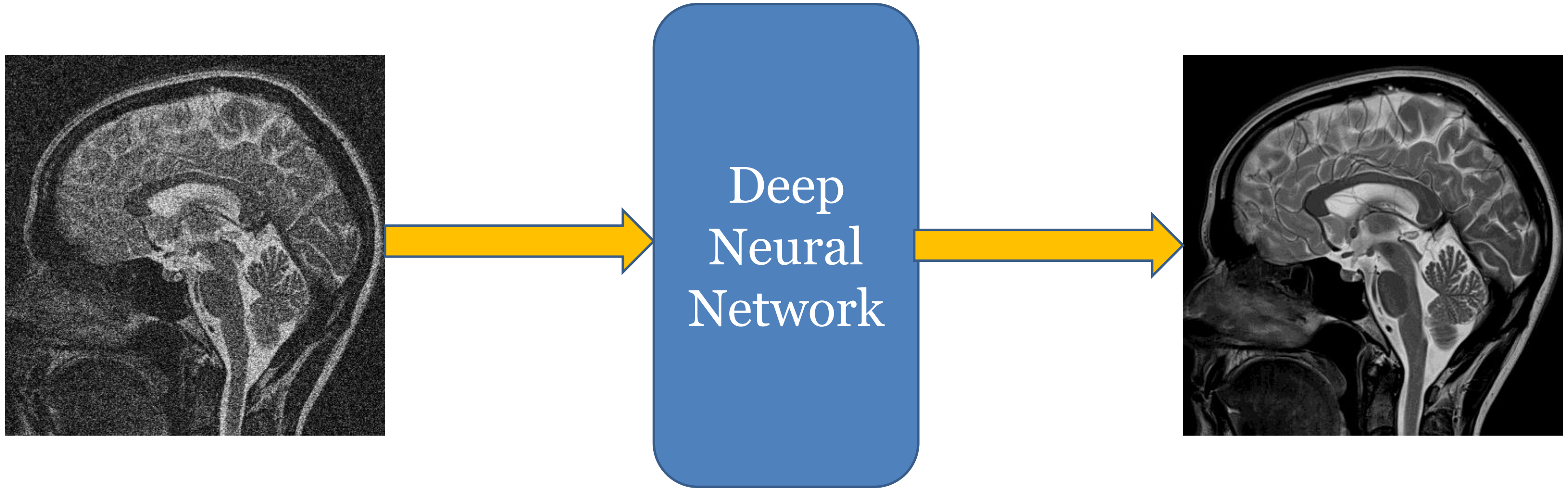
Deep Denosing

› There are two different strategy:

1. Input ($\boldsymbol{x}$): noisy image (or patch tensor), Output ($\boldsymbol{y}$): clean image
2. Input ($\boldsymbol{x}$): noisy image (or patch tensor), Output ($\boldsymbol{y}$): residual noise

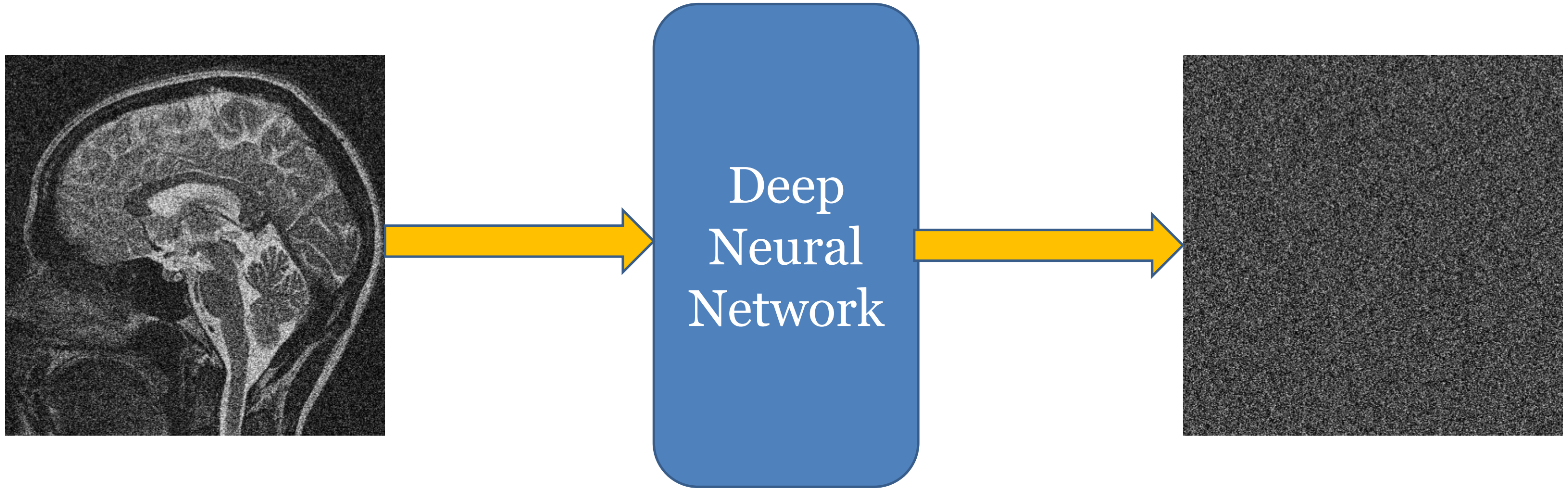
Deep Denosing

1. Deep Neural Network, predict clean image:



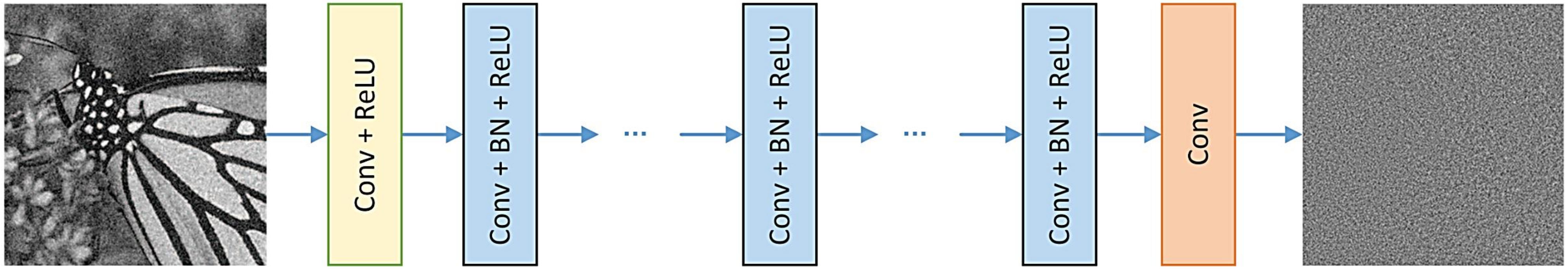
Deep Denosing

1. Deep Neural Network, predict residual noise:



DnCNN - Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising (IEEE, TIP, 2017)

1. DnCNN architecture:



- Noisy observation: $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$
- Deep CNN Mapping: $\widehat{\boldsymbol{n}} = \mathcal{R}(\boldsymbol{y}; \Theta)$
- Loss function: $l(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \|\mathcal{R}(\boldsymbol{y}_i; \Theta) - (\boldsymbol{y}_i - \boldsymbol{x}_i)\|_F^2 \longrightarrow \widehat{\boldsymbol{x}} = \boldsymbol{y} - \widehat{\boldsymbol{n}}$
- Where $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ is N noisy-clean training image (patch) pairs.

DnCNN - Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising (IEEE, TIP, 2017)

› DnCNN setting (1):

› Noise: AWGN

1. DnCNN-S (Known noise variance):
   - ❑Patch size: 40×40
   - ❑Number of training samples: 128×1600
   - ❑Noise variance: $\sigma = 15, 25, and\ 50$

DnCNN - Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising (IEEE, TIP, 2017)

› DnCNN setting (2):

› Noise: AWGN

1. DnCNN-B (Unknown noise variance):
   - ❑Patch size: 50×50
   - ❑Number of training samples: 128×3000
   - ❑Noise variance: $\sigma \in [0, 50]$

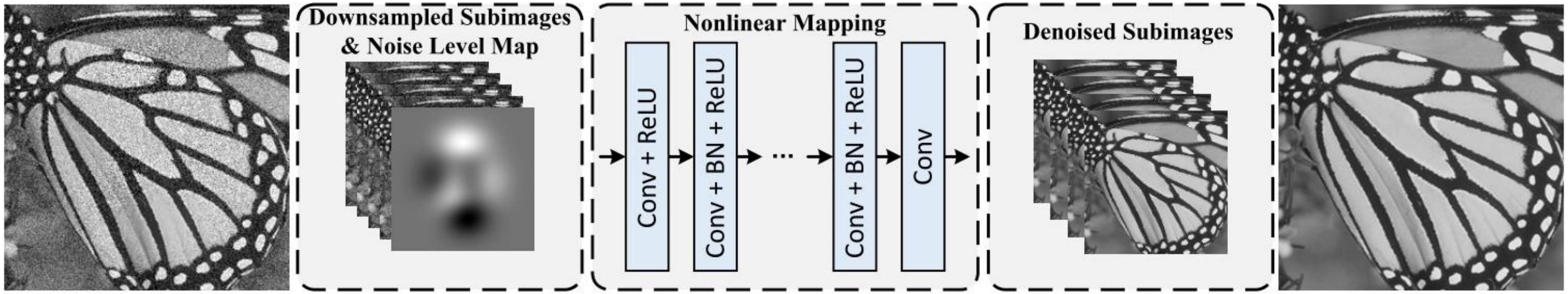
DnCNN - Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising (IEEE, TIP, 2017)

› Some results:

Methods	BM3D	WNNM	DnCNN-S	DnCNN-B
$\sigma = 15$	31.07	31.37	**31.73**	31.61
$\sigma = 25$	28.57	28.83	**29.23**	29.16
$\sigma = 50$	25.62	25.87	**26.23**	**26.23**

FFDNet - FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising (IEEE, TIP, 2018)

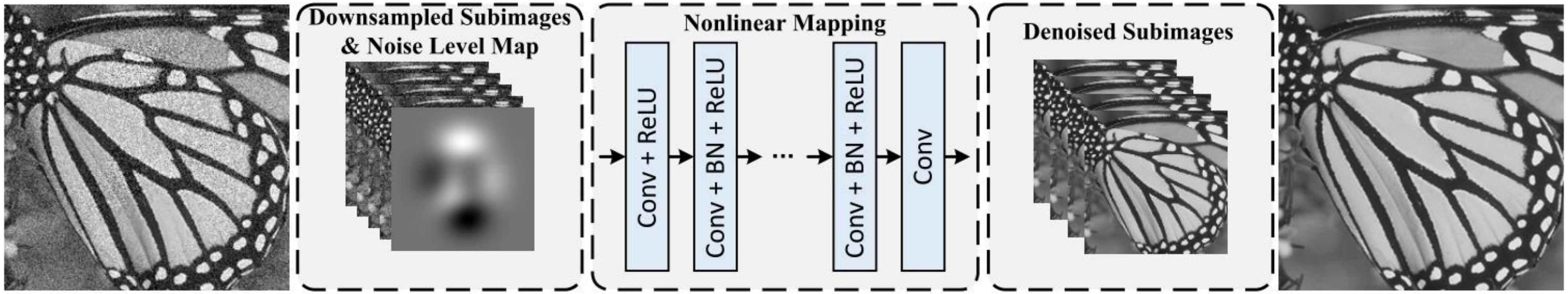
- FFDNet architecture:



- Noisy observation: $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$
- Deep CNN Mapping: $\hat{\boldsymbol{x}} = \mathcal{F}(\tilde{\boldsymbol{y}}_i, \boldsymbol{M}_i; \Theta)$
- Loss function: $l(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \|\mathcal{F}(\tilde{\boldsymbol{y}}_i, \boldsymbol{M}_i; \Theta) - \boldsymbol{x}_i\|_F^2$
- where $\{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i, \boldsymbol{M}_i)\}_{i=1}^{N}$ is N input-output training image (patch) pairs and $\boldsymbol{M}_i$ is noise level map.

FFDNet - FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising (IEEE, TIP, 2018)

› FFDNet architecture:



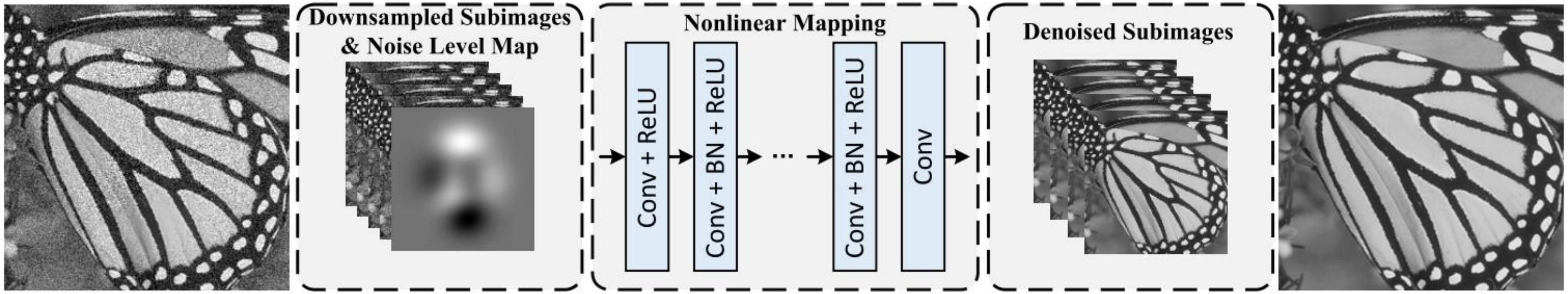
› Input tensor, $\tilde{\boldsymbol{y}}_i$, (assume image size is $W \times H$):

– Four down-sampled (reversible) sub-images of size $\frac{W}{2} \times \frac{H}{2}$

– Noise level map (For spatially invariant AWGN with noise level σ , $\boldsymbol{M}$ is a uniform map with all elements being σ) of size $\frac{W}{2} \times \frac{H}{2}$

FFDNet - FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising (IEEE, TIP, 2018)

› FFDNet architecture:



› Output, $\widehat{\boldsymbol{x}}$, is denoised image:

– After the last layer, an upscaling operation is applied as the *reverse operator* of the *down-sampling operator* applied in the input stage to produce the estimated clean image $\widehat{\boldsymbol{x}}$ of size $W \times H$.

FFDNet - FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising (IEEE, TIP, 2018)

› FFDNet setting:

› Noise: AWGN

- ❑Patch size: 70×70
- ❑Number of training samples: 128×8000
- ❑Noise variance: $\sigma \in [0, 75]$

FFDNet - FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising (IEEE, TIP, 2018)

› Some results

Methods	BM3D	WNNM	DnCNN	FFDNet
$\sigma = 15$	31.07	31.37	31.72	31.63
$\sigma = 25$	28.57	28.83	29.23	29.19
$\sigma = 35$	27.08	27.30	27.69	27.73
$\sigma = 50$	25.62	25.87	26.23	26.29
$\sigma = 75$	24.21	24.40	24.64	24.79

The End

› AnY QuEsTiOn?