Image Restoration: From Sparse and Low-rank Priors to Deep Priors

Lei Zhang
Dept. of Computing
The Hong Kong Polytechnic University
http://www.comp.polyu.edu.hk/~cslzhang

Outline

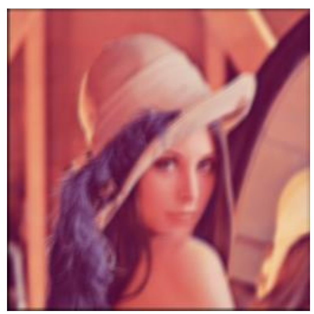
- Image restoration: the problem
- Sparse representation for image restoration
  - Sparse coding
  - Dictionary learning
  - Nonlocally centralized sparse representation
- Low-rank minimization for image restoration
  - Low-rank matrix approximation
  - Weighted nuclear norm minimization
- Deep learning for image restoration
  - Discriminative learning vs. model based optimization
  - Deep CNN methods for image restoration tasks
  - Learn a deep denoiser for general image restoration
- Open problems

Image restoration: the problem

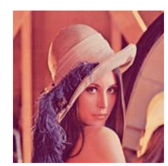
Image restoration: the problem

- Reconstruct the latent image from its degraded measurement
  - noise, down-sampling, blur, damaged pixels, …

Noisy

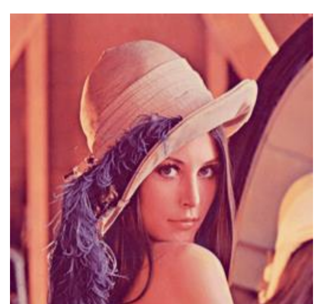


Blurred



Low-resolution

Damaged



General observation model

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v}$$

$\boldsymbol{H}$: The observation (degradation) matrix

$\boldsymbol{v}$: The additive noise

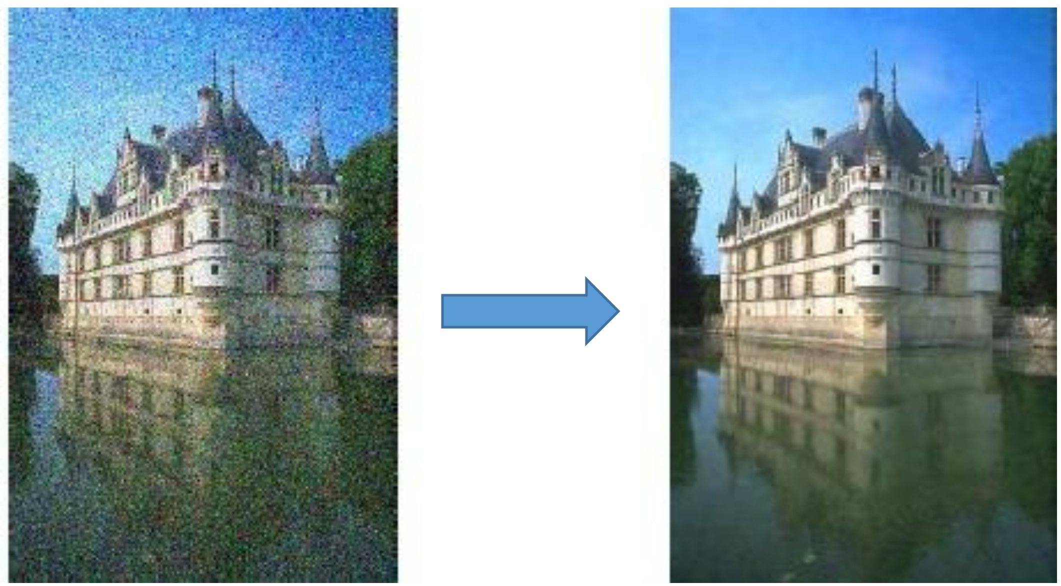
- Goal of image restoration:

  Given observation $\boldsymbol{y}$, recover the latent image $\boldsymbol{x}$.

- Image restoration is a typical ill-posed inverse problem. Prior information is needed to solve it.

Example applications

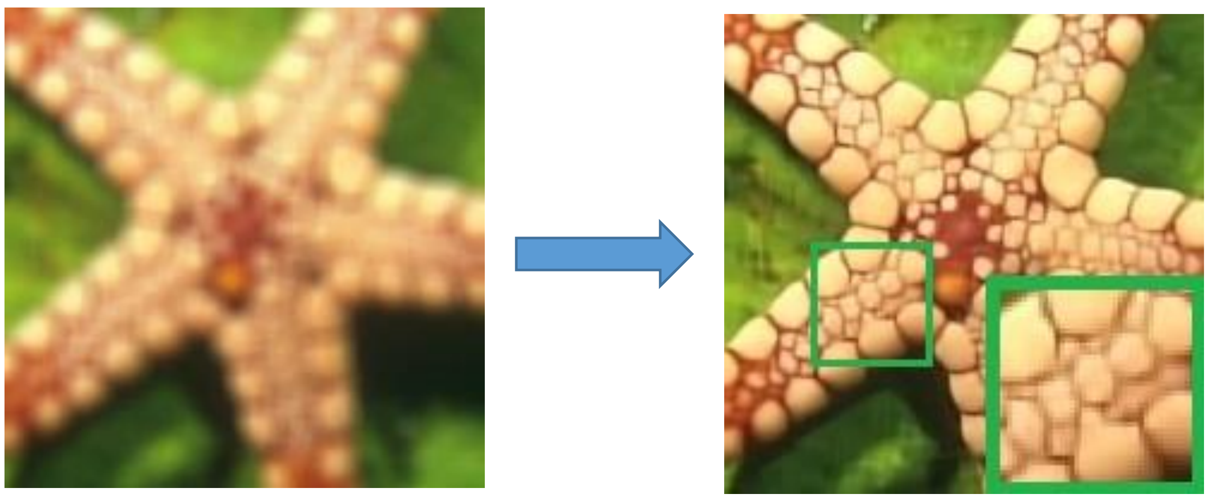
- Denoising



H is an identity matrix.

Example applications

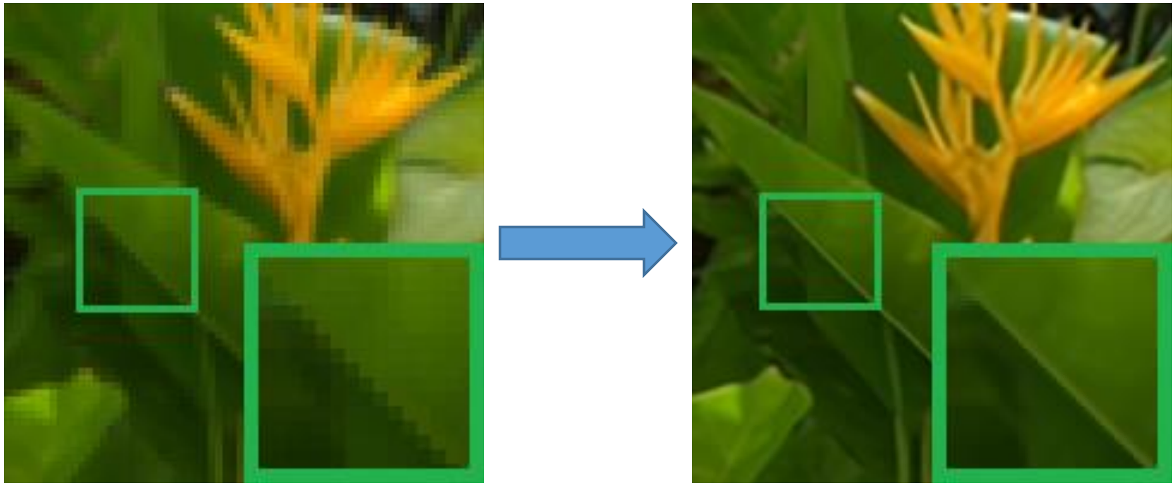
- Deblurring



H is a blurring matrix.

Example applications

- Superresolution



$\boldsymbol{H}$ is a compound matrix of
blurring and downsampling.

Example applications

- Inpainting

H is a 0-1 indication matrix
of damaged pixels.

Example applications

- Single image separation

$$y = x_1 + x_2 + e$$

Filtering based methods

- Gaussian low-pass filtering
  - Smoothing edges while removing noise
- PDE-based anisotropic diffusion
  - Preserving better edges than low-pass filtering
- Bilateral filtering
  - Exploiting both spatial and intensity similarity
- Nonlocal means filtering
  - Exploiting the nonlocal self-similarity

Local

Nonlocal

- From local filtering to nonlocal (global) filtering, the image restoration performance is greatly improved.

(Linear) Transform based methods

- Fourier transform ("big" sine and cosine wave bases)
- Wavelet transform ("small" and "localized" bases)
- Curvelet transform
  - More redundant, able to better describe big structures
- Ridgelet transform, Bandlet transform, …
  - More and more redundant, oriented, …
- The bases are actually the dictionary atoms.
- From Fourier dictionary to curvelet dictionary and so on, the dictionary becomes more and more redundant and over-complete.

Model based optimization

- Based on the image degradation process and the available image priors, build a model (objective function) and optimize it to estimate the latent image.
- General model:

Fidelity — Regularization (Prior)

$$min_{\boldsymbol{x}}\ F(\boldsymbol{x}, \boldsymbol{y}) + \lambda \cdot R(\boldsymbol{x})$$

- Many state-of-the-art methods belong to this category.
- Key issues
  - Modeling of the degradation process
  - Good priors about the latent image
  - Good objective function for minimization

Sparse representation for image restoration

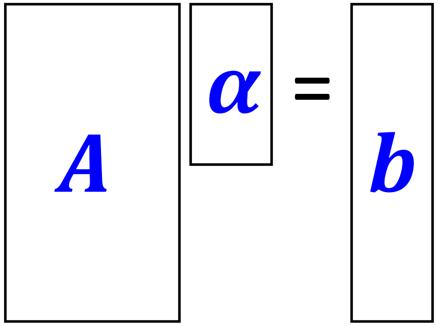
A linear system

$$\boldsymbol{A\alpha} = \boldsymbol{b}$$

- What is the solution $\boldsymbol{\alpha}$?
- It depends on the setting of matrix $\boldsymbol{A}$
  - If $\boldsymbol{A}$ is a full-rank square matrix, we have $\boldsymbol{\alpha} = \boldsymbol{A}^{-1}\boldsymbol{b}$.
  - If $\boldsymbol{A}$ is a full-rank but tall matrix (over-determined system), we can have an approximate solution by minimizing $||\boldsymbol{A\alpha} - \boldsymbol{b}||_2^2$.
  - We have:

$$\widehat{\boldsymbol{\alpha}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} = \boldsymbol{A}^{\dagger}\boldsymbol{b}$$

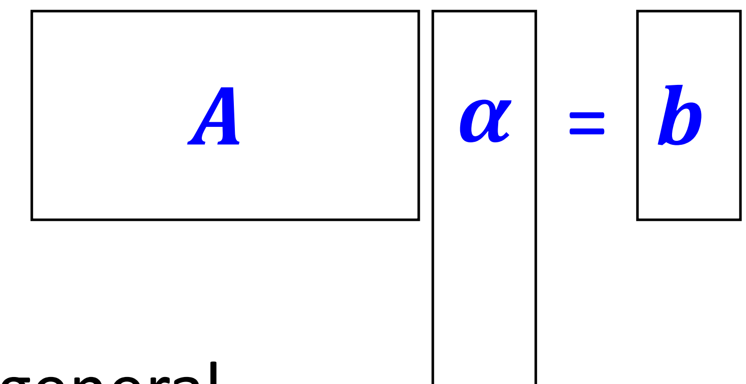
$\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$ is called the *pseudo-inverse* of $\boldsymbol{A}$.



Underdetermined linear system

$$\boldsymbol{A\alpha} = \boldsymbol{b}$$

- How about if $\boldsymbol{A}$ is a fat matrix (underdetermined system)?

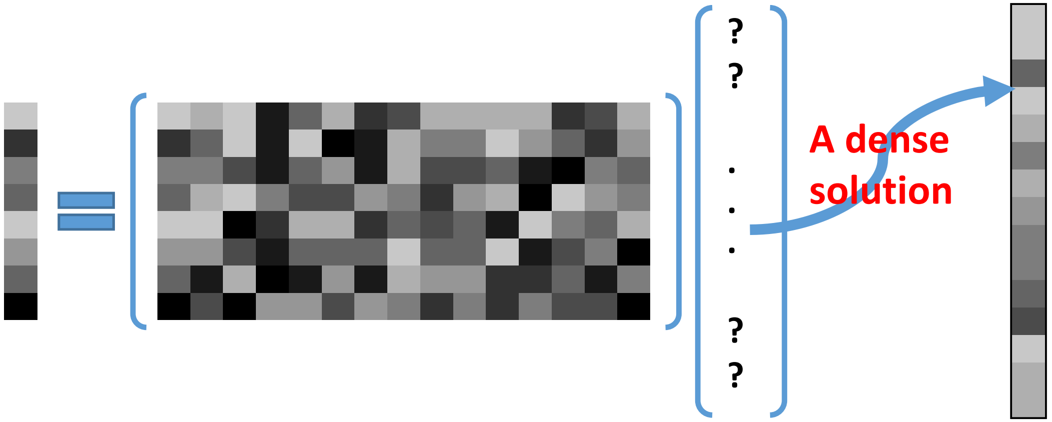


- There is no solution in general.
- Some constraint must be imposed to find a meaningful solution of $\boldsymbol{\alpha}$.

Solution

$$min_{\alpha} J(\boldsymbol{\alpha})\ s.t.\ \boldsymbol{A\alpha} = \boldsymbol{b}$$

- Different objective functions $J(\boldsymbol{\alpha})$ lead to different solutions to the underdetermined system.
- A dense solution: $J(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_2^2$

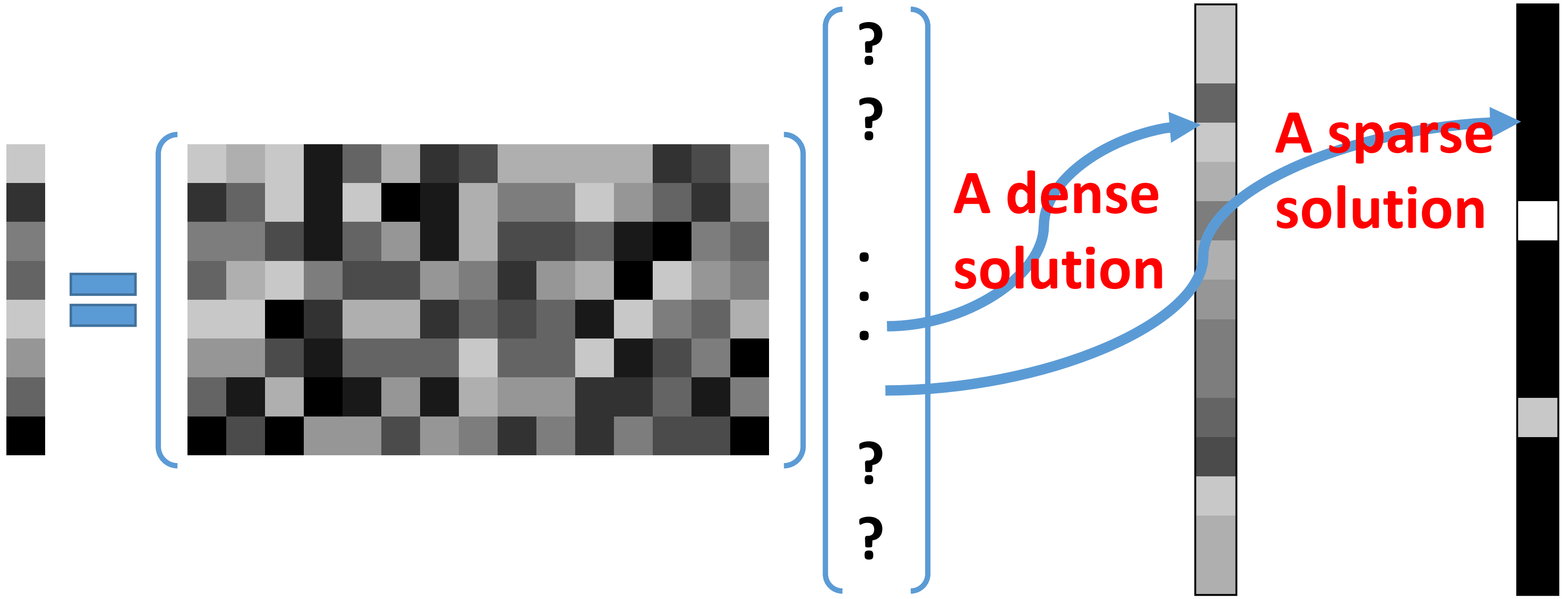


Sparse solution

- The dense solution may not be useful or effective enough (e.g., not robust, not unique).
- In many applications, we may need a "sparse" solution that has many zero or nearly zero entries (e.g., more robust, more unique).
- So how to achieve this goal?

A model for sparse solution

$$min_{\alpha} \|\boldsymbol{\alpha}\|_0 \; s.t. \boldsymbol{A\alpha} = \boldsymbol{b}$$



A convex model

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \; s.t. \; \boldsymbol{A\alpha} = \boldsymbol{b}$$

L_0-norm minimization is non-convex and NP-hard.

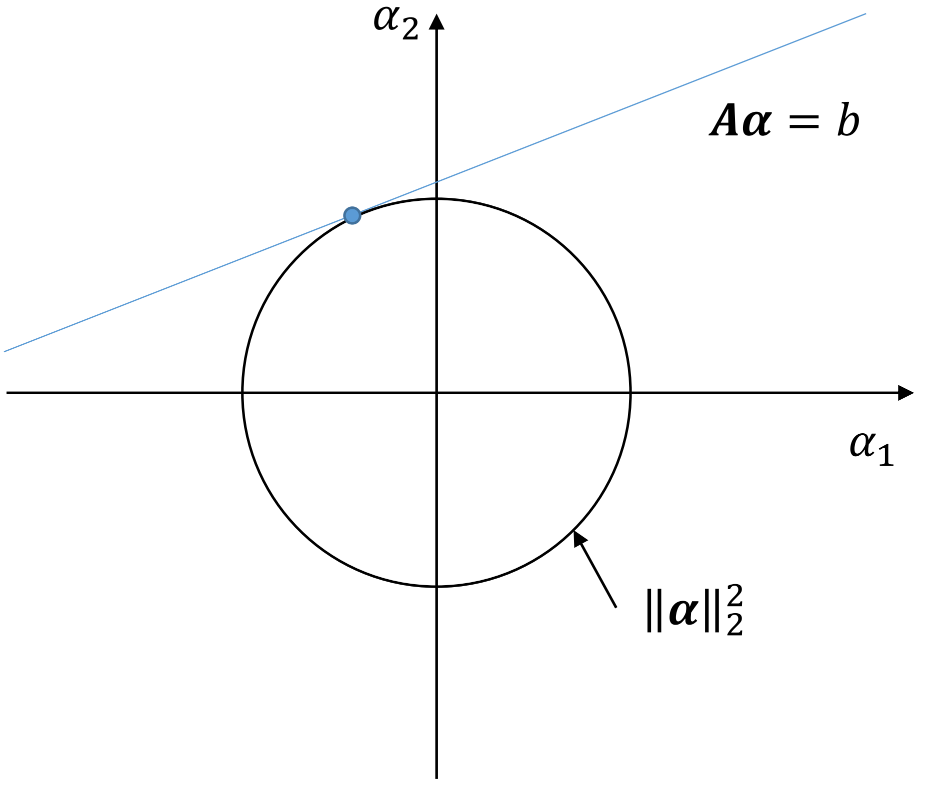
$$\downarrow$$

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \; s.t. \; \boldsymbol{A\alpha} = \boldsymbol{b}$$

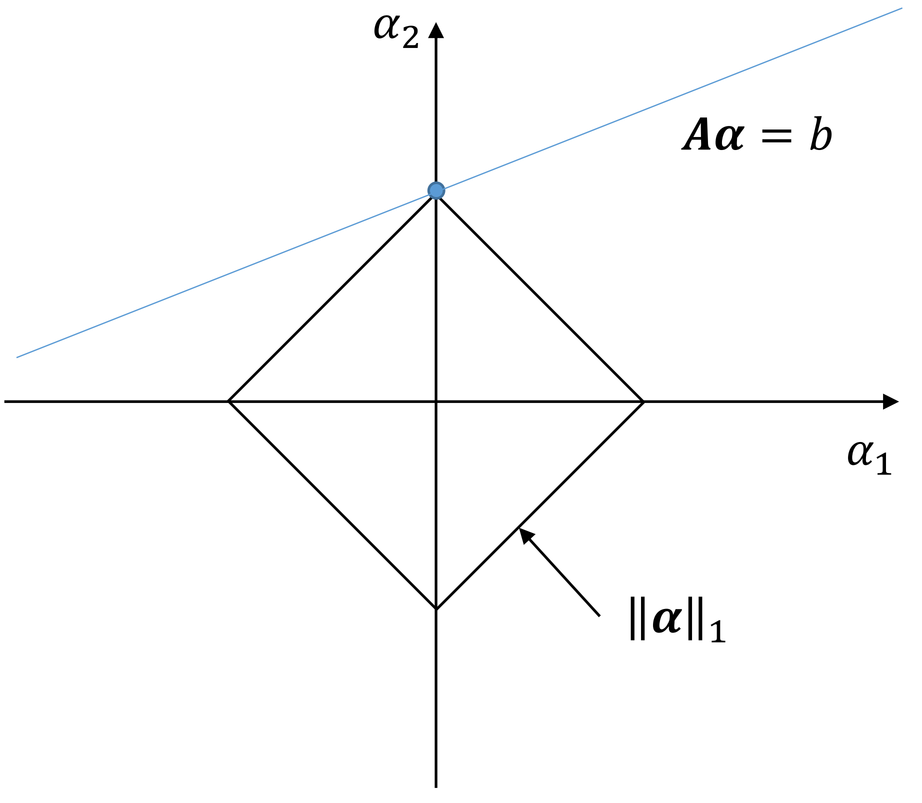
L_1-norm minimization is tightest convex relaxation of L_0-norm minimization .

L_2-norm vs. L_1-norm

- Geometric illustration



$min_{\alpha}\|\boldsymbol{\alpha}\|_2^2 \; s.t. \boldsymbol{A\alpha} = \boldsymbol{b}$

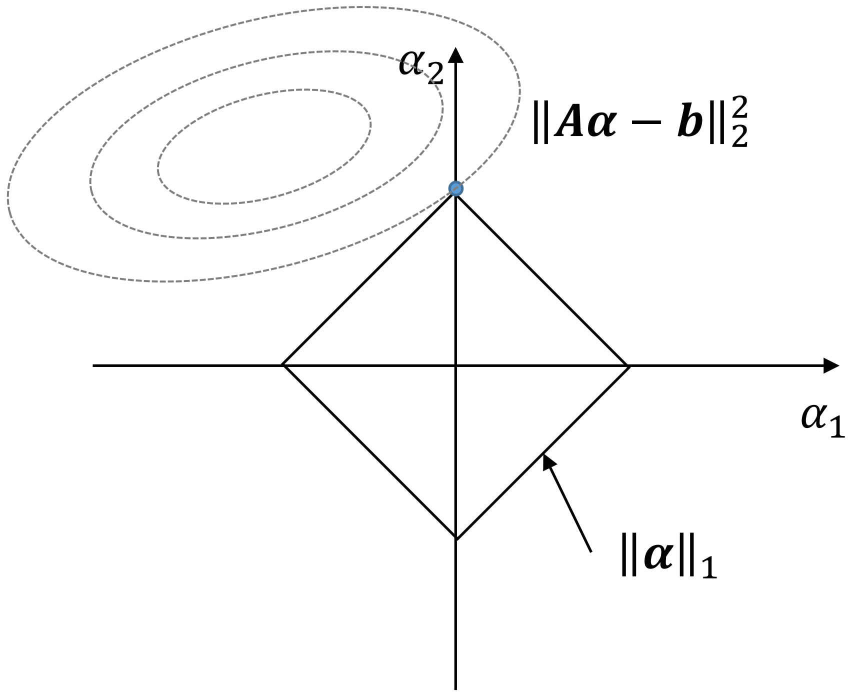


$min_{\alpha}\|\boldsymbol{\alpha}\|_1 s.t. \boldsymbol{A\alpha} = \boldsymbol{b}$

A relaxed L_1 sparse coding model

$$min_{\boldsymbol{\alpha}} \|\boldsymbol{A\alpha} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

- This is the most widely used sparse coding model, which is easy to solve and usually leads to a sparse solution.



Sparse coding solvers

- Greedy Search for L_0-norm minimization
  - Matching pursuit (MP)
  - Orthogonal matching pursuit (OMP)
- Convex Optimization for L_1-norm minimization
  - Linear programming
  - Iteratively reweighted least squares
  - Proximal gradient descent (Iterative soft-thresholding)
  - Augmented Lagrangian methods (Alternating direction method of multipliers)

How to adopt sparse coding for image restoration?

- Represent (encode) $\boldsymbol{x}$ over a dictionary $\boldsymbol{D}$, while enforcing the representation vector to be sparse:

$$min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \; s.t. \; \boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$$

- Together with $min_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + R(\boldsymbol{x})$, we have:

$$min_{\boldsymbol{\alpha}} \|\boldsymbol{H}\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

- Solving $\boldsymbol{x}$ turns to solving $\boldsymbol{\alpha}$.

Sparse representation based Image restoration: basic procedures

1. Partition the degraded image into overlapped patches.
2. For each patch, solve the following nonlinear L_1-norm sparse coding problem:

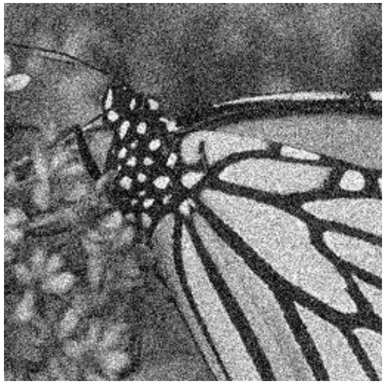
$$\hat{\boldsymbol{\alpha}} = argmin_{\boldsymbol{\alpha}} \; \|\boldsymbol{HD\alpha} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1$$

3. Reconstruct each patch by $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\alpha}}$.
4. Put the reconstructed patch back to the original image. For overlapped pixels between patches, average them.
5. In practice, the above procedures can be iterated for several rounds to better reconstruct the image.

An example

- A noisy image and the denoised images in several iterations

Noisy Image



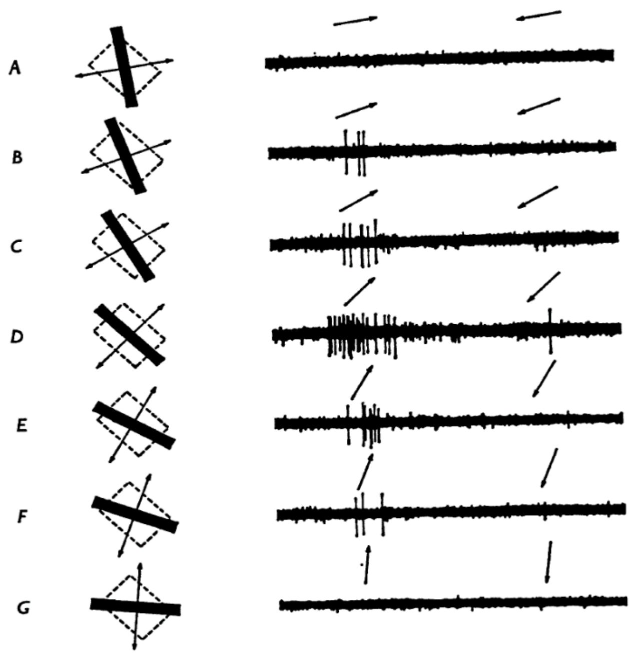
Iter 1

Iter 3

Iter 5

Why sparse: neuroscience perspective

- Observations on Primary Visual Cortex
  - *The Monkey Experiment by Hubel and Wiesel, 1968*



Responses of a simple cell in monkeys' right striate cortex.

David Hubel and Torsten Wiesel
Nobel Prize Winner

Why sparse: neuroscience perspective

- Olshausen and Field's Sparse Codes, 1996
  - Goal: to achieve a coding strategy that succeeds in producing full set of natural images while keeping all the three properties: *localized*, *oriented* and *bandpass*.
  - Solution: a coding strategy that maximizes sparseness:

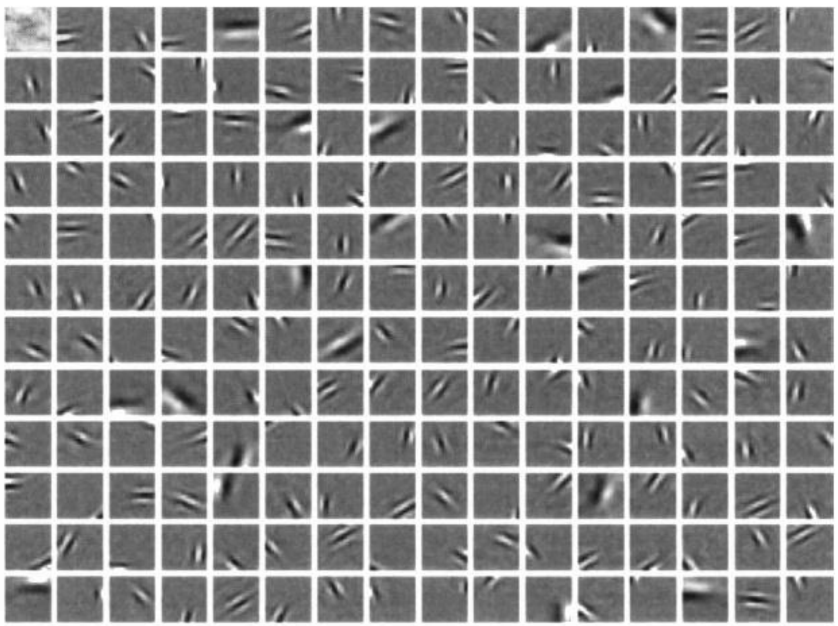
E = -[preserve information] – lambda×[sparseness]

- Bruno A. Olshausen, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images." Nature, 381.6583 (1996): 607-609.
- Bruno A. Olshausen and David J. Field. "Sparse coding with an overcomplete basis set: A strategy employed by VI?." Vision Research, 37.23 (1997): 3311-3326.

Why sparse: neuroscience perspective

- Olshausen and Field's Sparse Codes, 1996
  - The basis function can be updated by gradient descent:

$$\Delta\phi_i(x_m, y_n) = \eta\left\langle a_i\left[I(x_m, y_n) - \hat{I}(x_m, y_n)\right]\right\rangle$$



Resulted basis functions.

Courtesy by Olshausen and Field, 1996

Why sparse: Bayesian perspective

- Signal recovery in a Bayesian viewpoint

Likelihood **Prior**

$$\hat{\boldsymbol{x}} = argmax_{\boldsymbol{x}}\, P(\boldsymbol{x}|\boldsymbol{y}) \propto argmax_{\boldsymbol{x}}\, P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x})$$

  - Encode $\boldsymbol{x}$ over a dictionary $\boldsymbol{D}$

$$\boldsymbol{x} = \boldsymbol{D\alpha}$$

  - Assume that the representation coefficients follow some exponential distribution (prior):

$$\boldsymbol{\alpha} \sim exp\left(-\sum_i \|\alpha_i\|_p\right)$$

Why sparse: Bayesian perspective

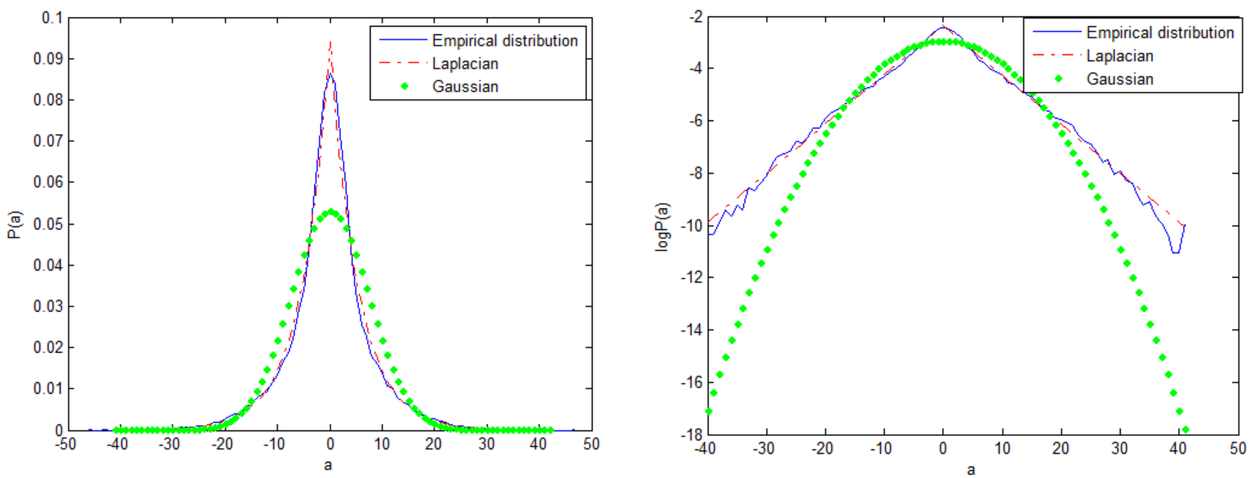
- The maximum a posteriori (MAP) solution:

$$\hat{\boldsymbol{\alpha}} = argmax_{\alpha}\, P(\boldsymbol{\alpha}|\boldsymbol{y})$$
$$= argmax_{\alpha} -log\, P(\boldsymbol{y}|\boldsymbol{\alpha}) - log P(\boldsymbol{\alpha})$$
$$= argmin_{\alpha}\, ||\boldsymbol{HD\alpha} - \boldsymbol{y}||_2^2 + \lambda||\boldsymbol{\alpha}||_p$$

- We can see:
  - If $p = 0$, it is the L_0-norm sparse coding problem.
  - If $p = 1$, it becomes the convex L_1-norm sparse coding problem.
  - If $0 < p < 1$, it will be the non-convex L_p-norm minimization.

Why sparse: Bayesian perspective

- Is $p \leq 1$ a good prior of $\boldsymbol{\alpha}$? In general, yes!

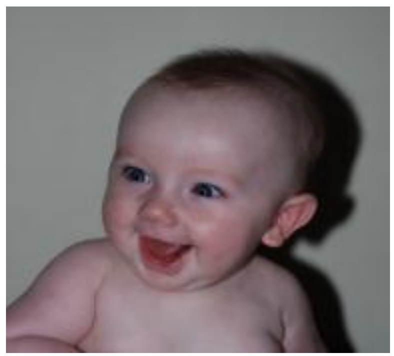


- Empirical distribution of image coding coefficients on an over-complete dictionary. (Right: log-probability)
  - L_1-norm minimization: MAP with Laplacian prior.
  - L_2-norm minimization: MAP with Gaussian prior.

Why sparse: signal processing perspective

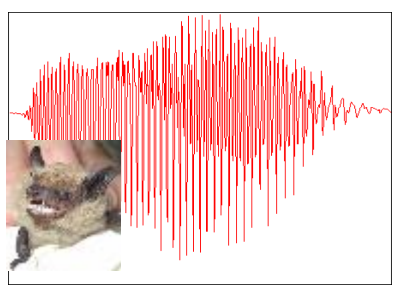
- Some examples:

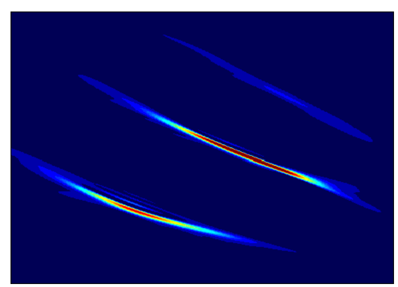
N pixels



$K \ll N$ large wavelet coefficients (blue = 0)

N wideband signal samples





$K \ll N$ large Gabor (TF) coefficients

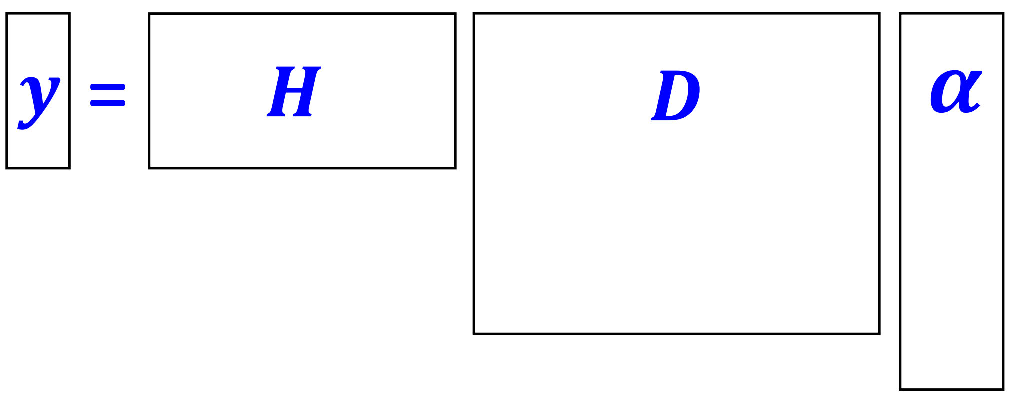
Courtesy by Baraniuk, 2012

Why sparse: signal processing perspective

- K-sparse signal: $\boldsymbol{x}$ is called K-sparse if it is a linear combination of only K basis vectors. If $K \ll N$, it is called compressible.

$$\boldsymbol{x} = \sum_{i=1}^{K} \alpha_i \boldsymbol{\psi}_i = \boldsymbol{D}\boldsymbol{\alpha}$$

- Measurement $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} = \boldsymbol{H}\boldsymbol{D}\boldsymbol{\alpha} = \boldsymbol{A}\boldsymbol{\alpha}$



Why sparse: signal processing perspective

- Reconstruction
  - If $\boldsymbol{x}$ is K-sparse, it is possible that we can reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$ with a number of measurements much less than the signal dimension ($M \ll N$):

$$\widehat{\boldsymbol{\alpha}} = argmin_{\boldsymbol{\alpha}} \, \|\boldsymbol{\alpha}\|_0 \; s.t. \, \boldsymbol{y} = \boldsymbol{A}\boldsymbol{\alpha}$$

  - But the measurement matrix $\boldsymbol{A}$ should satisfy the **RIP** condition.

Why sparsity helps signal recovery?

❖An illustrative example

- You are looking for your another half.
  - i.e., you are "reconstructing" the desired signal.
- You hope that she/he is "白-富-美"/ "高-富-帅".
  - i.e., you want a "clean" and "perfect" reconstruction.
- However, there are limited candidates.
  - i.e., the dictionary is small. (For example, your search space is constrained to a class in PolyU ☹.)

- Can you easily find your ideal another half?

Why sparsity helps signal recovery?

❖A illustrative example

- Candidate A is tall; however, he is too poor.
- Candidate B is rich; however, he is too fat.
- Candidate C is handsome; however, he is not healthy.
- If you sparsely select one of them, none is ideal for you
  - i.e., a sparse representation vector such as [0, 1, 0].
- How about a dense solution: (A+B+C)/3?
  - i.e., a dense representation vector [1, 1, 1]/3
- The "reconstructed one" is somewhat "高-富-帅", but he is fat and unhealthy (i.e., noise) at the same time.

Why sparsity helps signal recovery?

❖A illustrative example

- So what’s the problem?
  - This is because the dictionary is too small!
- If you are able to find your another half from all candidates all over the world (i.e., a large enough dictionary) , there is a very high probability (nearly 1) that you will find the one.
  - i.e., a very sparse solution [0, …, 1, …, 0].
- In summary, a sparse solution with an over-complete dictionary often works!
- Sparsity (coefficients) and redundancy (dictionary) are the two sides of the same coin.

Dictionary

- Analytical dictionaries
  - DCT bases
  - Wavelets
  - Curvelets
  - Ridgelets, bandlets, …
- Learn dictionaries from natural images
  - K-SVD
  - Coordinate descent
  - Multi-scale dictionary learning
  - Adaptive PCA dictionaries
  - …

Why dictionary learning?

- Sparse models with a learned over-complete dictionary often work better than analytically designed dictionaries such as DCT dictionary and wavelet dictionary.
- Why learned dictionary works better?
  - More adaptive to specific task/data.
  - Less strict constraints on the mathematical properties of basis (dictionary atom).
  - More flexible to model data.
  - Tend to produce sparser solutions to many problems.

Dictionary learning methods

- Input: Training samples $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_n]$
- Output: Dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \dots, \boldsymbol{d}_m], m < n$, such that $\boldsymbol{Y} \approx \boldsymbol{D}\boldsymbol{\Lambda}$, and $\boldsymbol{\Lambda} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_n]$

- Methods
  - K-SVD (L_0-norm)
  - Coordinate descent (L_1-norm)
  - Others
    - Multiscale dictionary learning
    - Double sparsity dictionary learning
    - Adaptive PCA dictionary learning

K-SVD

- Basic idea
  - K-means is a special case of sparse dictionary learning (approximate each sample with only one atom, i.e., the cluster center). The idea of alternatively updating cluster label and cluster center in k-means can be adopted for dictionary learning.
  - Instead of approximating each sample using only one atom, we can learn a dictionary of K atoms to approximate a sample:

$$min_{\boldsymbol{\alpha},\boldsymbol{D}} \sum_j \left\| \boldsymbol{D}\boldsymbol{\alpha}_j - \boldsymbol{y}_j \right\|_2^2, s.t. \forall j, \left\| \boldsymbol{\alpha}_j \right\|_0 \leq L$$

  - Since L_0-norm is adopted, when updating $\boldsymbol{D}$, we only care about the number of non-zeros in $\boldsymbol{\alpha}$ but not the values of them.

- M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing, 54 (11), 4311-4322.

K-SVD

- Algorithm
  - The coding phase can be solved by conventional sparse coding algorithms, such as MP, OMP, et al.

$$\widehat{\boldsymbol{\alpha}_j} = argmin_{\boldsymbol{\alpha}_j} \left\|\boldsymbol{\alpha}_j\right\|_0 \; s.t. \, \boldsymbol{y}_j = \boldsymbol{D}\boldsymbol{\alpha}_j$$

  - For the dictionary updating phase, K-SVD update dictionary column by column:

$$\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{\Lambda}\|_{\boldsymbol{F}}^2 = \left\|\boldsymbol{Y} - \sum_{k=1}^{K} \boldsymbol{d}_k \boldsymbol{\alpha}_k\right\|_{\boldsymbol{F}}^2$$

$$= \left\|\left(\boldsymbol{Y} - \sum_{i \neq k}^{K} \boldsymbol{d}_i \boldsymbol{\alpha}_i\right) - \boldsymbol{d}_k \boldsymbol{\alpha}_k\right\|_{\boldsymbol{F}}^{\boldsymbol{2}} = \|\boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{\alpha}_k\|_{\boldsymbol{F}}^{\boldsymbol{2}}$$

Only select non-zeros in $\boldsymbol{\alpha}_k$ to update corresponding $\boldsymbol{d}_k$

$$\|\boldsymbol{E}_k \boldsymbol{\Omega}_k - \boldsymbol{d}_k \boldsymbol{\alpha}_k \boldsymbol{\Omega}_k\|_F^2$$

Rank 1 approximation of matrix

- Code available at: http://www.cs.technion.ac.il/~elad/software/

L_1 dictionary learning

- Basic idea and algorithm
  - Inspired by K-SVD, L_1 dictionary learning and adopt the same strategy of alternatively updating dictionary and coefficients.
  - Different from L_0 norm which only cares about the number of non-zeros, the coefficient value is also important in L_1 norm.

$$min_{\boldsymbol{D},\boldsymbol{\Lambda}} \|\boldsymbol{D}\boldsymbol{\Lambda} - \boldsymbol{Y}\|_F^2 + \lambda \|\boldsymbol{\Lambda}\|_{\boldsymbol{1}}, s.t. \left\|\boldsymbol{d_j}\right\|_2^2 \leq 1$$

$$min_{\boldsymbol{\Lambda}} \|\boldsymbol{D}\boldsymbol{\Lambda} - \boldsymbol{Y}\|_F^2 + \lambda \|\boldsymbol{\Lambda}\|_{\boldsymbol{1}}$$

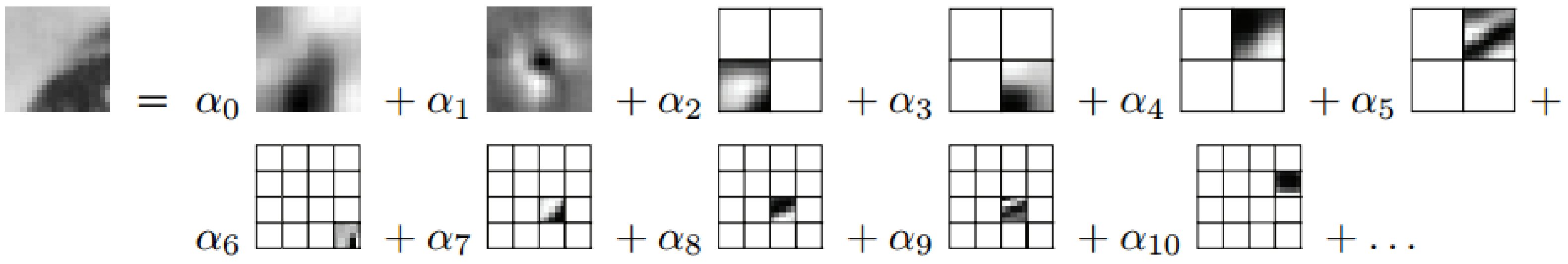
$$min_{\boldsymbol{D}} \|\boldsymbol{D}\boldsymbol{\Lambda} - \boldsymbol{Y}\|_F^2, s.t. \left\|\boldsymbol{d_j}\right\|_2^2 \leq 1$$

- Meng Yang, et. al. "Metaface Learning for Sparse Representation based Face Recognition," In ICIP 2010. (Code: http://www4.comp.polyu.edu.hk/~cslzhang/code/ICIP10/Metaface_ICIP.rar)

Multi-scale dictionary learning

- Motivation

  The complexity of sparse coding increases exponentially with signal dimension. Therefore, most methods work on small image patches. To perform sparse coding on larger patches, multi-scale method can provide a way to adaptively model simple structure in larger scales and details in smaller scales.



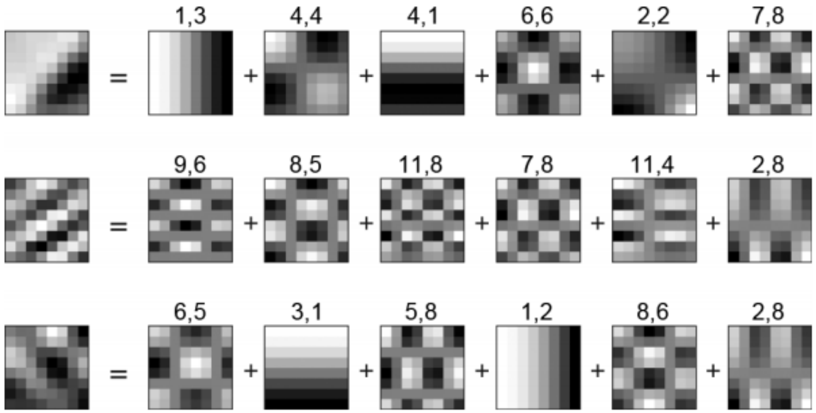
- J. Mairal, et al., Learning multiscale sparse representations for image and video restoration. Multiscale Modeling & Simulation.

Double sparsity: sparse dictionary learning for high dimensional data

- Learn a sparse dictionary
  - To model high-dimensional (e.g. large patch for image) data, we can require the dictionary is sparse too.
  - Double sparsity models the dictionary to be learned as $\boldsymbol{D} = \boldsymbol{\Phi Z}$, where $\boldsymbol{\Phi}$ is some pre-defined bases, such as DCT or wavelets.
  - The objective function of double sparsity model is:

$$min_{\boldsymbol{\Phi},\boldsymbol{Z},\boldsymbol{\Lambda}} \sum \|\boldsymbol{Y} - \boldsymbol{\Phi Z \Lambda}\|_F^2 \,,$$

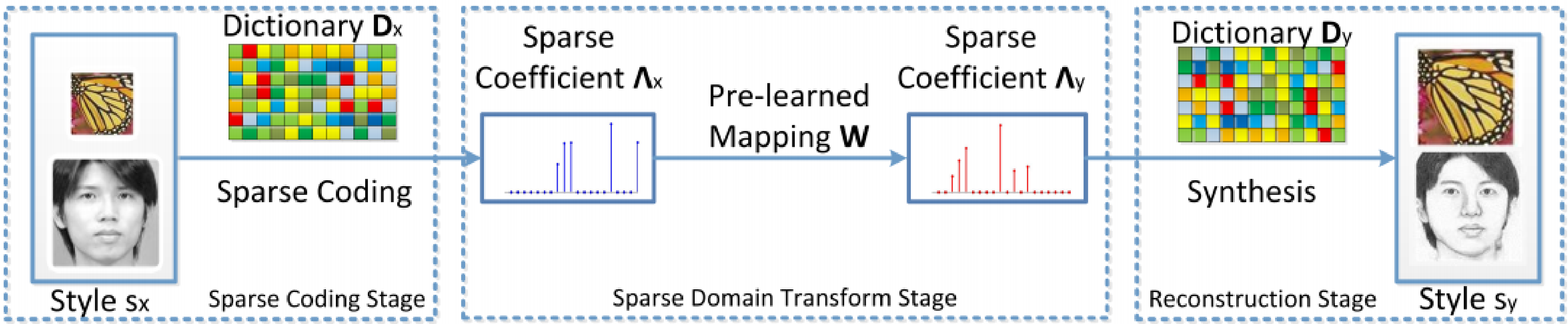
$$s.t. \forall j, \|\boldsymbol{\alpha}_j\|_0 \le L, \|\boldsymbol{z}_j\|_0 = K$$



- R. Rubinstein, et. al. Double Sparsity: Learning Sparse Dictionaries for Sparse Signal Approximation. IEEE Trans. on Signal Processing, 2010.

Semi-coupled dictionary learning

- Flexible to model complex image structures

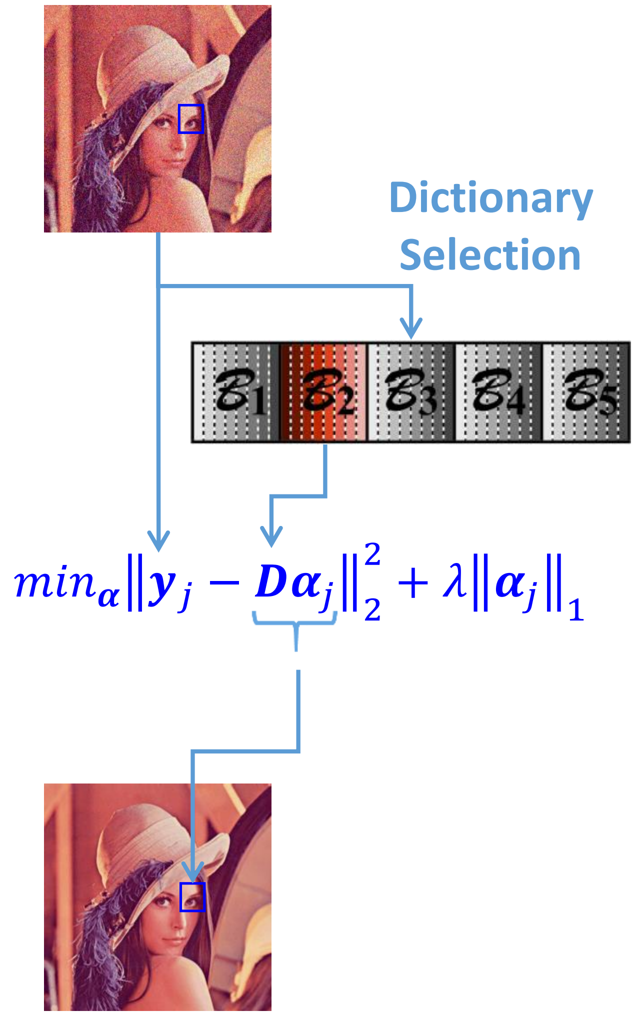


$$\min_{\{\mathbf{D}_x,\mathbf{D}_y,\mathbf{W}\}} \|\mathbf{X}-\mathbf{D}_x\mathbf{\Lambda}_x\|_F^2 + \|\mathbf{Y}-\mathbf{D}_y\mathbf{\Lambda}_y\|_F^2$$
$$+\gamma\|\mathbf{\Lambda}_y-\mathbf{W}\mathbf{\Lambda}_x\|_F^2 + \lambda_x\|\mathbf{\Lambda}_x\|_1 + \lambda_y\|\mathbf{\Lambda}_y\|_1 + \lambda_W\|\mathbf{W}\|_F^2$$
$$\text{s.t. } \|\mathbf{d}_{x,i}\|_{l_2} \le 1, \|\mathbf{d}_{y,i}\|_{l_2} \le 1, \forall i$$

- S. Wang, L. Zhang, Y. Liang, Q. Pan, "Semi-Coupled Dictionary Learning with Applications to Image Super-Resolution and Photo-Sketch Image Synthesis," In CVPR 2012.
- http://www4.comp.polyu.edu.hk/~cslzhang/SCDL/SCDL_Code.zip

Adaptive PCA dictionary selection

- Motivation
  - Sparse coding is time consuming, especially with large dictionaries.
  - A large over-complete dictionary is often required to model complex image local structures.
  - We can learn a set of PCA dictionaries, and select one of them to represent a given image patch.



- W. Dong, L. Zhang, G. Shi, X. Wu, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, TIP 2011.
- http://www4.comp.polyu.edu.hk/~cslzhang/ASDS_data/TIP_ASDS_IR.zip

Image nonlocal self-similarity prior

- In natural images, usually we can find many similar patches to a given path, which may be spatially far from it. This is called image nonlocal self-similarity.
- Nonlocal self-similarity has been widely and successfully used in image restoration.

- A. Buades, et al., A non-local algorithm for image denoising. CVPR 2005.

Non-locally centralized sparse representation (NCSR)

- A neat but very effective sparse representation model, which naturally integrates nonlocal self-similarity (NSS) prior and sparse coding.

W. Dong, L. Zhang and G. Shi, "Centralized Sparse Representation for Image Restoration", in ICCV 2011.

W. Dong, L. Zhang, G. Shi and X. Li, "Nonlocally Centralized Sparse Representation for Image Restoration", IEEE Trans. on Image Processing, vol. 22, no. 4, pp. 1620-1630, April 2013.

http://www4.comp.polyu.edu.hk/~cslzhang/code/NCSR.rar

NCSR: The idea

- For true signal

$$\boldsymbol{\alpha_x} = argmin_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1, s.t.\ \|\boldsymbol{x} - \boldsymbol{D\alpha}\|_2^2 \leq \varepsilon$$

- For degraded signal

$$\boldsymbol{\alpha_y} = argmin_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1, s.t.\ \|\boldsymbol{y} - \boldsymbol{HD\alpha}\|_2^2 \leq \varepsilon$$

- The sparse coding noise (SCN)

$$\boldsymbol{\upsilon_\alpha} = \boldsymbol{\alpha_y} - \boldsymbol{\alpha_x}$$

- To better reconstruct $\boldsymbol{x}$, we should reduce the SCN $\boldsymbol{\upsilon_\alpha}$:

$$\tilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \boldsymbol{x} \approx \boldsymbol{D\alpha_y} - \boldsymbol{D\alpha_x} = \boldsymbol{D\upsilon_\alpha}$$

NCSR: The objective function

- The proposed objective function

$$\boldsymbol{\alpha}_{\boldsymbol{y}} = argmin_{\boldsymbol{\alpha}}\left\{\|\boldsymbol{y} - \boldsymbol{HD\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}}_{\boldsymbol{x}}\|_p\right\}$$

  - Key idea: Suppressing the SCN

- How to compute $\widehat{\boldsymbol{\alpha}}_{\boldsymbol{x}}$?
  - The unbiased estimate: $\widehat{\boldsymbol{\alpha}}_{\boldsymbol{x}} = E[\boldsymbol{\alpha}_{\boldsymbol{x}}]$
  - The zero-mean property of SCN $\boldsymbol{\upsilon}_{\boldsymbol{\alpha}}$ makes

$$\widehat{\boldsymbol{\alpha}}_{\boldsymbol{x}} = E[\boldsymbol{\alpha}_{\boldsymbol{x}}] \approx E[\boldsymbol{\alpha}_{\boldsymbol{y}}]$$

NCSR: The solution

- The nonlocal estimation of $E[\boldsymbol{\alpha}_{\boldsymbol{y}}]$

$$\boldsymbol{\mu}_i = \sum_{j \in C_i} \omega_{i,j} \boldsymbol{\alpha}_{i,j}, \ \omega_{i,j} = exp\left(\left\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_{i,j}\right\|_2^2 / h\right) / W$$

- The simplified objective function

$$\boldsymbol{\alpha}_{\boldsymbol{y}} = argmin_{\boldsymbol{\alpha}} \left\{ \|\boldsymbol{y} - \boldsymbol{HD\alpha}\|_2^2 + \lambda \sum_{i=1}^{N} \|\boldsymbol{\alpha}_i - \boldsymbol{\mu}_i\|_p \right\}$$

- The iterative solution:

$$\boldsymbol{\alpha}_i^{(n)} = argmin_{\boldsymbol{\alpha}} \left\{ \|\boldsymbol{y} - \boldsymbol{HD\alpha}\|_2^2 + \lambda \sum_{i=1}^{N} \left\|\boldsymbol{\alpha}_i - \boldsymbol{\mu}_i^{(n-1)}\right\|_p \right\}$$

NSCR: The parameters and dictionaries

- The L_p-norm is set to L_1-norm since SCN is generally Laplacian distributed.
- The regularization parameter λ is adaptively determined based on the MAP estimation principle.
- Local adaptive PCA dictionaries are used, which are learned from the given image.
  - Cluster the image patches, and for each cluster, a PCA dictionary is learned and used to code the patches within this cluster.

Low-rank minimization for image restoration

Motivation

- Visual data often has an intrinsic low-rank structure

Face images

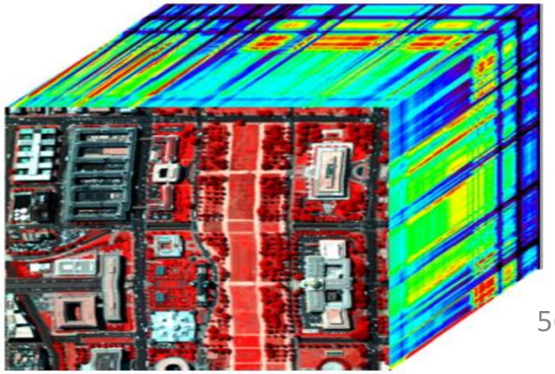
Well aligned face images lie on a low-dimensional subspace.

Surveillance video

Video background of a static scene is always of very low-rank structure.

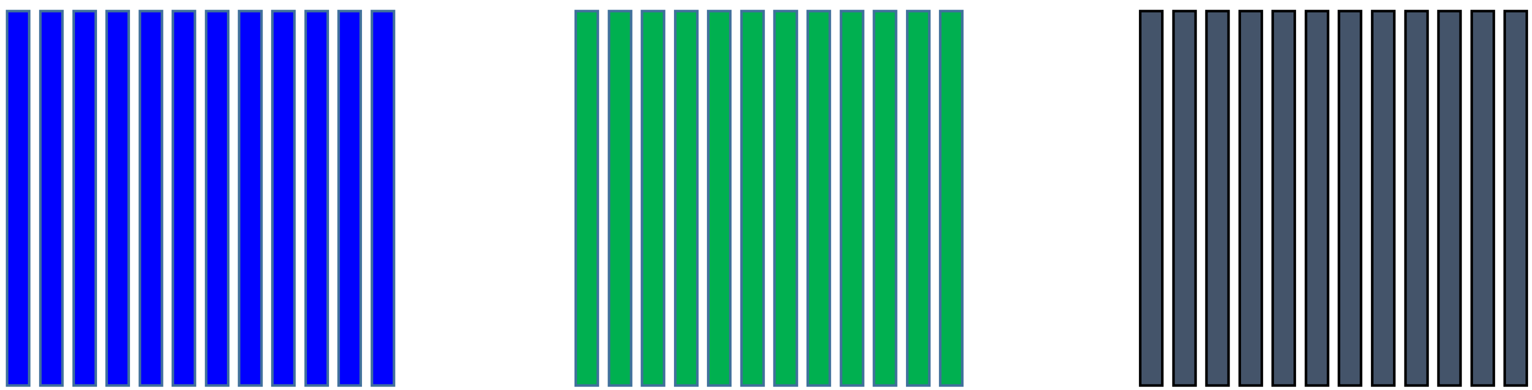
Multispectral image

Different bands of a multi-spectral image are highly correlated, holding a low-rank property along spectrum



Data representation

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$$



Each column corresponds to a sample

The desired latent low-rank matrix

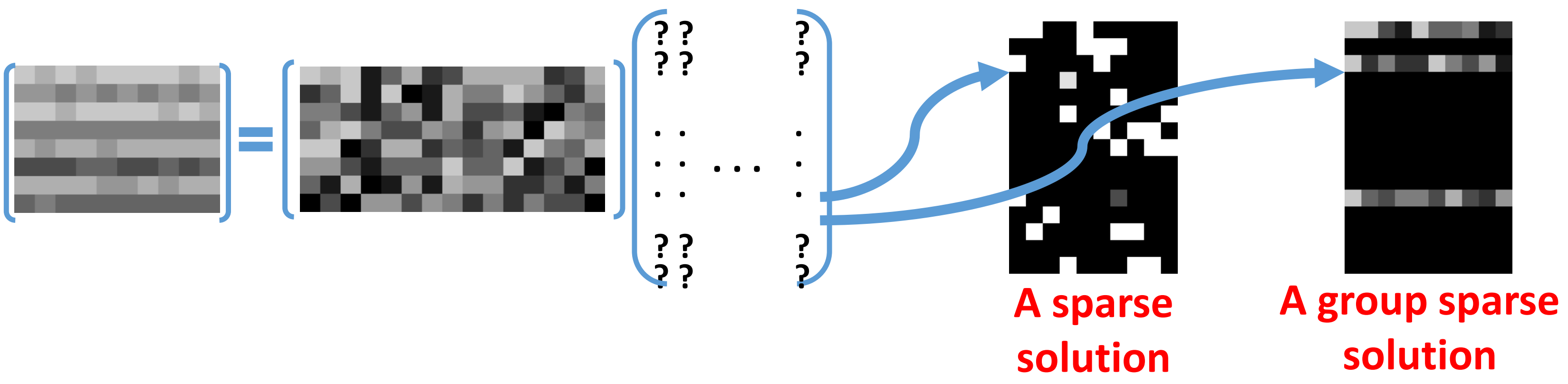
The residual matrix

Group sparsity

- How to characterize the sparsity of a group of correlated vectors?
- Group sparsity:

$$min_{\boldsymbol{\Lambda}} J(\boldsymbol{\Lambda}) \quad s.t. \boldsymbol{X} = \boldsymbol{D}\boldsymbol{\Lambda}$$

- Group sparsity is still a kind of 1D encoding.



From 1D to 2D: rank minimization

- Considering the fact (i.e., prior) that the input vectors are highly correlated, we can take them as a 2D low rank matrix and minimize its rank:

$$Rank(\boldsymbol{X}) = \sum \|\sigma_i(\boldsymbol{X})\|_0$$

- Rank minimization represents the input matrix over a set of rank 1 basis matrices.
- However, minimization of $Rank(\boldsymbol{X})$ is non-convex and NP hard!

Nuclear norm

$$Rank(\boldsymbol{X}) = \sum \|\sigma_i(\boldsymbol{X})\|_0$$

- The above rank function is non-convex. A convex relaxation of it is the so-called nuclear norm:

$$\|\boldsymbol{X}\|_* = \sum \|\sigma_i(\boldsymbol{X})\|_1$$

Nuclear norm minimization

- Nuclear norm minimization (NNM) can be used to estimate the latent low rank matrix $\boldsymbol{X}$ form $\boldsymbol{Y}$ via the following unconstrained minimization problem:

$$\hat{\boldsymbol{X}} = argmin_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_*$$

- Closed form solution (Cai, et al., SIAM10)

$$\hat{\boldsymbol{X}} = \boldsymbol{U} S_\lambda(\boldsymbol{\Sigma}) \boldsymbol{V}^T$$

where $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of $\boldsymbol{Y}$, and

$$S_\lambda(\boldsymbol{\Sigma})_{ii} = \max\left(\boldsymbol{\Sigma}_{ii} - \frac{\lambda}{2}, 0\right)$$

- J.-F. Cai, E.J. Candès and Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optimiz.*, 20(4): 1956--1982, 2010.

NNM: *pros* and *cons*

$$\hat{\boldsymbol{X}} = argmin_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_*$$
$$\hat{\boldsymbol{X}} = \boldsymbol{U} S_\lambda(\boldsymbol{\Sigma}) \boldsymbol{V}^T$$

- Pros
  - ✓Tightest convex envelope of rank minimization.
  - ✓Closed form solution.
- Cons
  - × Treat equally all the singular values, ignoring the different significances of matrix singular values.

Weighted nuclear norm minimization (WNNM)

- Weighted nuclear norm

$$\|\boldsymbol{X}\|_{w,*} = \sum \|w_i \sigma_i(\boldsymbol{X})\|_1$$

- WNNM model

$$\hat{\boldsymbol{X}} = argmin_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \|\boldsymbol{X}\|_{w,*}$$

- Difficulties
  - The WNNM is not convex for general weight vectors

Optimization of WNNM

Theorem 1. $\forall \boldsymbol{Y} \in R^{m\times n}$, let $\boldsymbol{Y} = \boldsymbol{U\Sigma V}^T$ be its SVD. The optimal solution of the WNNM problem:

$$\widehat{\boldsymbol{X}} = argmin_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \|\boldsymbol{X}\|_{w,*}$$

is

$$\widehat{\boldsymbol{X}} = \boldsymbol{UDV}^T$$

where $\boldsymbol{D}$ is a diagonal matrix with diagonal entries $\boldsymbol{d}$=$[d_1, d_2, \dots, d_r]$ $(r = min(m, n))$ determined by:

$$min_{d_1,\, d_2 \dots d_n} \sum_{i=1}^{r} (d_i - \sigma_i)^2 + w_i\, d_i$$
$$s.t.\, d_1 \geq d_2 \geq d_r \geq 0. \quad \blacksquare$$

- S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, “Weighted Nuclear Norm Minimization and Its Applications to Low Level Vision,” International Journal of Computer Vision, 2017.

An important corollary

Corollary 1. If the weights satisfy $0 \leq w_1 \leq w_2 \leq w_n$, the non-convex WNNM problem has a closed form optimal solution:

$$\hat{\boldsymbol{X}} = \boldsymbol{U} S_w(\boldsymbol{\Sigma}) \boldsymbol{V}^T$$

where $\boldsymbol{Y} = \boldsymbol{U\Sigma V}^T$ is the SVD of $\boldsymbol{Y}$, and

$$S_w(\boldsymbol{\Sigma})_{ii} = max\left(\boldsymbol{\Sigma}_{ii} - \frac{w_i}{2}, 0\right). \blacksquare$$

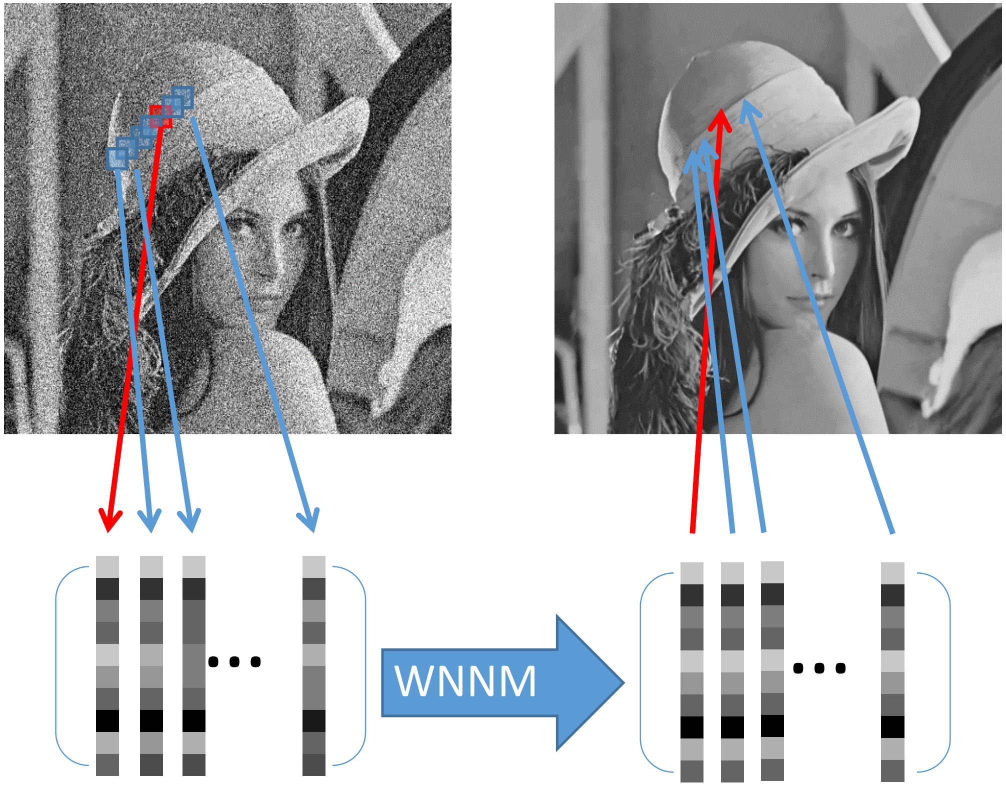
- S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, "Weighted Nuclear Norm Minimization and Its Applications to Low Level Vision," International Journal of Computer Vision, 2017.

Application of WNNM to image denoising

$$\hat{\boldsymbol{X}} = argmin_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \|\boldsymbol{X}\|_{w,*}$$

1) For each noisy patch, search in the image for its nonlocal similar patches to form matrix $\boldsymbol{Y}$.
2) Solve the WNNM problem to estimate the clean patches $\boldsymbol{X}$ from $\boldsymbol{Y}$.
3) Put the clean patch back to the image.
4) Repeat the above procedures several times to obtain the denoised image.

WNNM based image denoising



- S. Gu, L. Zhang, W. Zuo and X. Feng, "Weighted Nuclear Norm Minimization with Application to Image Denoising," CVPR 2014.

The weights

- Model

$$\widehat{\boldsymbol{X}} = argmin_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \|\boldsymbol{X}\|_{w,*}$$

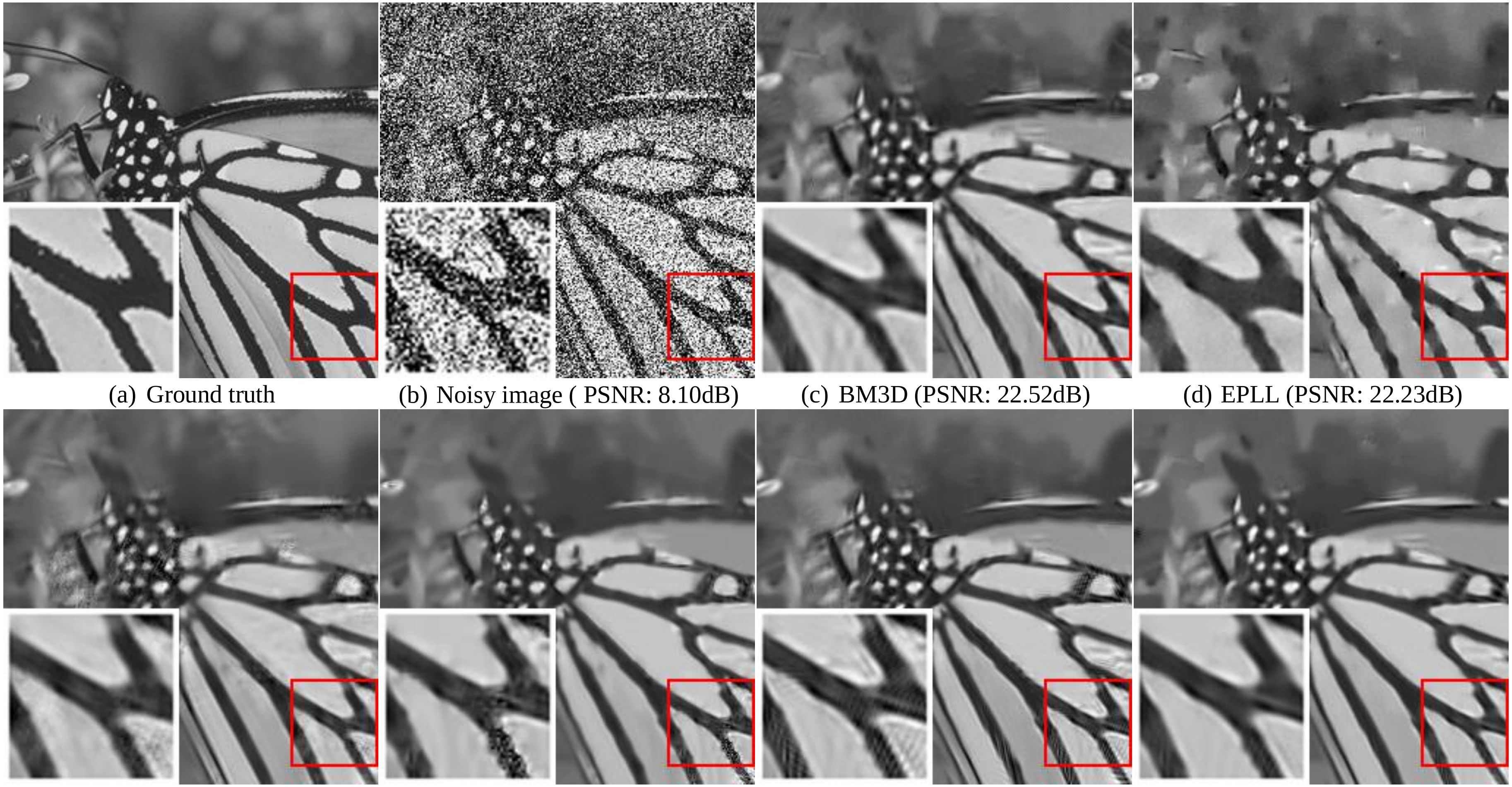
- Weights

$$w_i = \frac{c\sqrt{n}}{\hat{\sigma}_i(\boldsymbol{X}) + \varepsilon}$$

where

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

Experimental results



(a) Ground truth (b) Noisy image (PSNR: 8.10dB) (c) BM3D (PSNR: 22.52dB) (d) EPLL (PSNR: 22.23dB)

(e) SSC (PSNR: 22.24dB) (f) NCSR (PSNR: 22.11dB) (g) SAIST (PSNR: 22.61dB) (h) WNNM (PSNR: 22.91dB)

Denoising results on image *Monarch* by different method (noise level sigma=100).

Robust PCA (RPCA)

- In some applications, the residual $\boldsymbol{E} = \boldsymbol{Y} - \boldsymbol{X}$ may not be Gaussian distributed or may be sparse, then $\|\boldsymbol{Y} - \boldsymbol{X}\|_F^2$ will not be a good way to model residual.
- The L_1–norm is more robust to characterize sparse errors. We have the following robust PCA (RPCA) model:

$$min_{\boldsymbol{X}} \|\boldsymbol{X}\|_* + \|\boldsymbol{X} - \boldsymbol{Y}\|_1$$

$$\Downarrow$$

$$min_{\boldsymbol{X}} \|\boldsymbol{X}\|_* + \|\boldsymbol{E}\|_1$$
$$s.t. \;\; \boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$$

Extension of WNNM to RPCA

- The objective function:

$$min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{w,*} + \|\boldsymbol{E}\|_1$$
$$s.t.\ \ \boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$$

- We can use the ALM method to solve it:

$$L(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{Y}, \mu) = \|\boldsymbol{X}\|_{w,*} + \left\|\boldsymbol{E}\right\|_1 + \langle \boldsymbol{Y}, \boldsymbol{D} - \boldsymbol{A} - \boldsymbol{E} \rangle + \frac{\mu}{2} \left\|\boldsymbol{D} - \boldsymbol{A} - \boldsymbol{E}\right\|_F^2$$

Background modeling

RPCA-WNNM

Original video

Background

Foreground

RPCA-NNM

Extension to matrix completion

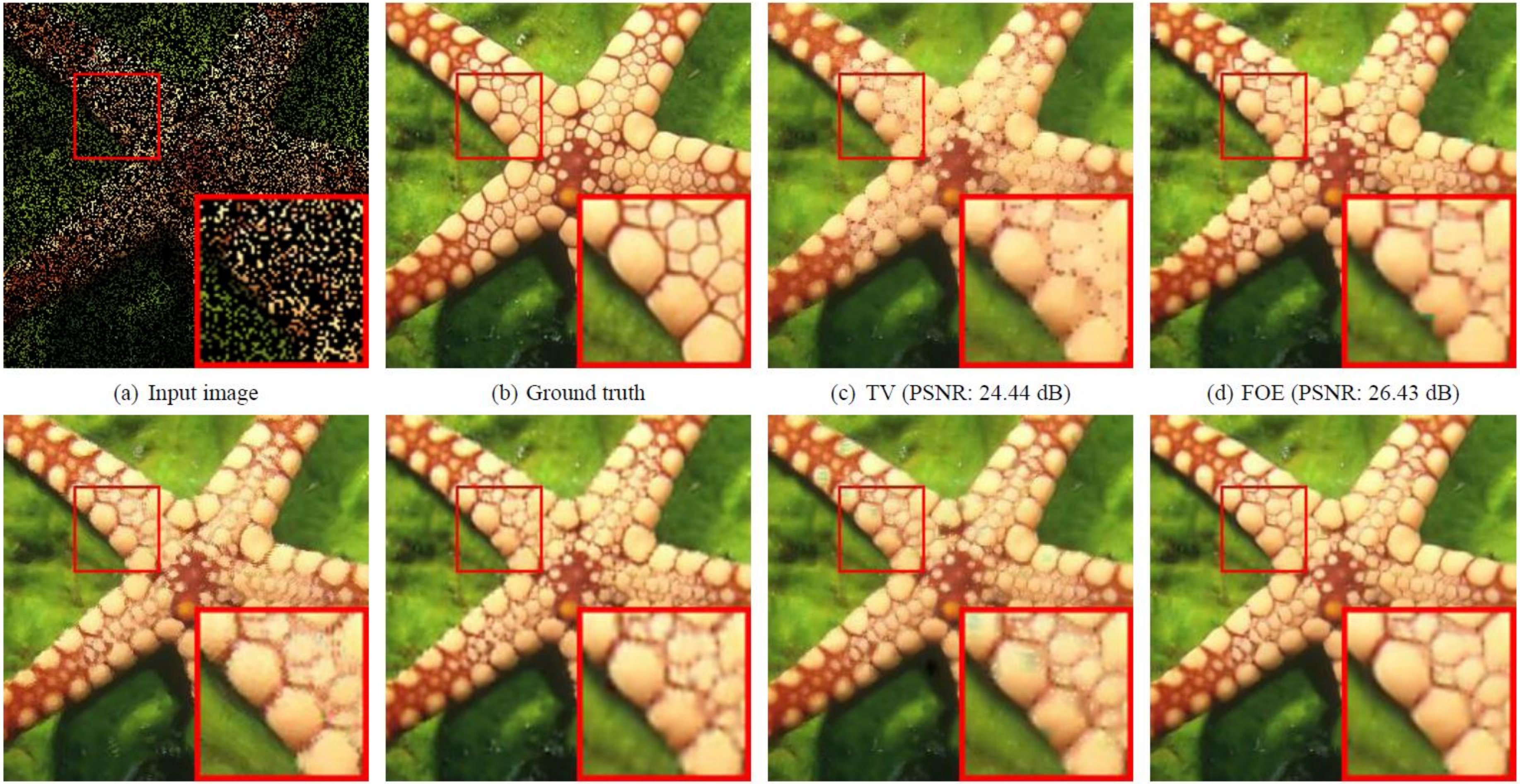
- The objective function:

$$min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{w,*}$$
$$s.t. P_{\Omega}(\boldsymbol{X}) = P_{\Omega}(\boldsymbol{Y})$$

- We can use the ALM method to solve it:

$$\Gamma = \|\boldsymbol{X}\|_{w,*} + \left\|\boldsymbol{E}\right\|_1 + \langle \boldsymbol{L}, \boldsymbol{Y} - \boldsymbol{X} - \boldsymbol{E} \rangle$$
$$+ \frac{\mu}{2} \left\|\boldsymbol{Y} - \boldsymbol{X} - \boldsymbol{E}\right\|_F^2$$

Image inpainting



(a) Input image (b) Ground truth (c) TV (PSNR: 24.44 dB) (d) FOE (PSNR: 26.43 dB)

(e) VNL (PSNR: 24.36 dB) (f) BPDL (PSNR: 26.57 dB) (g) NNM (PSNR: 25.45 dB) (h) WNNM (PSNR: 27.11 dB)

Deep learning for image restoration

Discriminative learning for image restoration

- Learn a compact inference or a mapping function from a training set of degraded-latent image pairs.
- General formulation:

Loss function — Set of parameters to be learned

$$min_{\Theta} loss(\hat{\boldsymbol{x}}, \boldsymbol{x}) \quad s.t. \hat{\boldsymbol{x}} = F(\boldsymbol{y}, \boldsymbol{H}; \Theta)$$

- Key issues
  - The availability of paired training data
  - The design of learning architecture
  - The definition of loss function

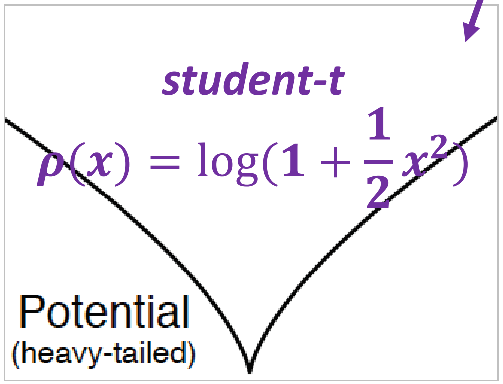
Shrinkage fields

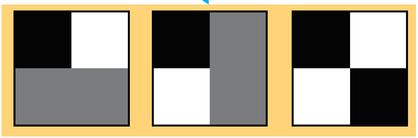
$$\hat{\boldsymbol{x}} = argmin_{\boldsymbol{x}} \frac{\lambda}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \sum_{i=1}^{N} \sum_{c \in C} \rho_i(\mathrm{f}_i^{\mathrm{T}} \boldsymbol{x}_{(c)})$$

Cliques

Potential function e.g., student-t, GMM

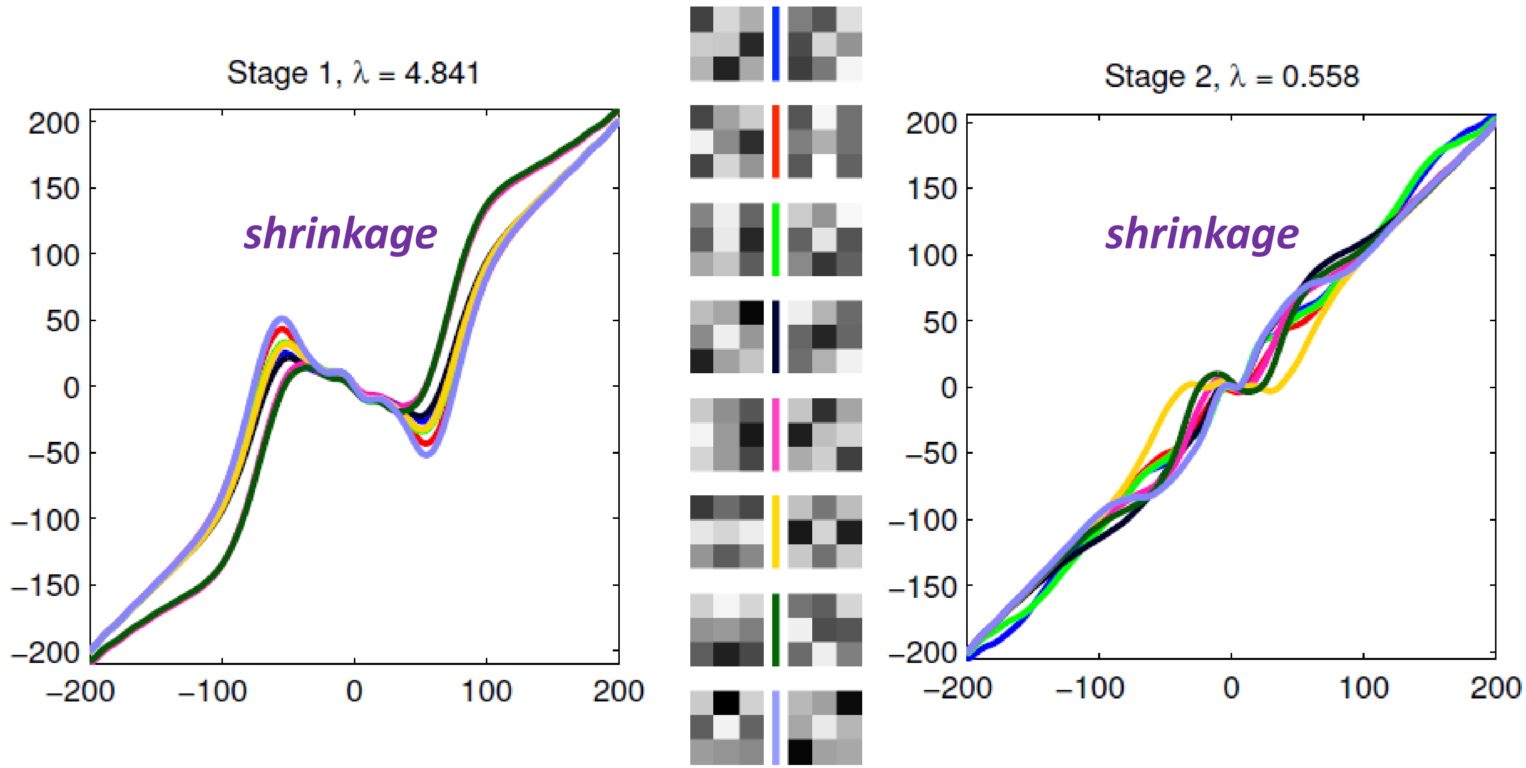
N convolution filters, e.g.,





U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In CVPR 2014

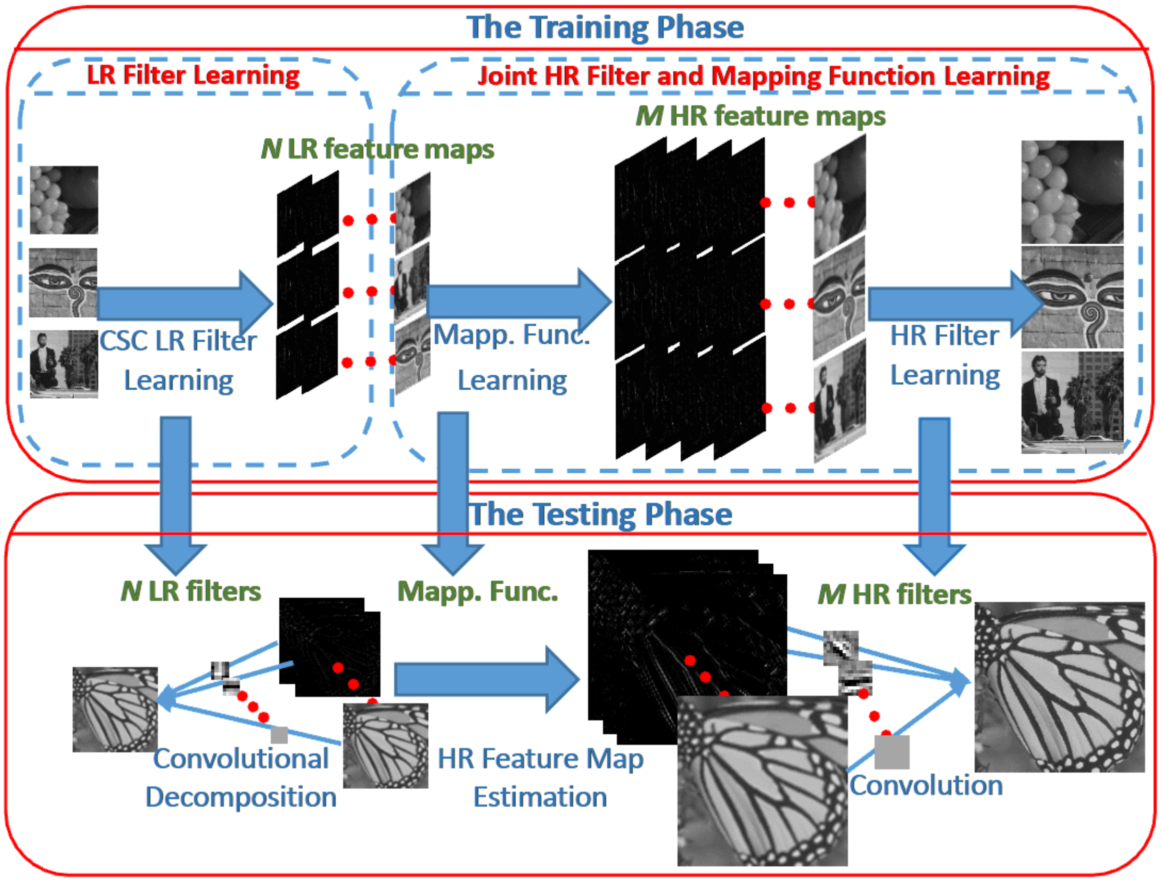
Shrinkage fields



Shrinkage functions are not limited to monotonic functions.

First two stages of learned csf3×3 model. The shrinkage functions are color-matched with their corresponding filters.

Convolutional sparse coding for image super-resolution



S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, L. Zhang, "Convolutional Sparse Coding for Image Super-resolution," in ICCV 2015.

Model based optimization vs. discriminative learning

- Model based optimization methods
  - ✓General to handle different image restoration problems
  - ✓Clear physical meaning
  - × The hand-crafted prior may not be strong enough
  - × The optimization process can be time consuming
- Discriminative learning based methods
  - ✓Data driven end-to-end learning
  - ✓Can be very efficient in the testing stage
  - × The generality of learned models is limited
  - × The interpretability of learned models is limited

Why deep learning?

- Strong learning capacity
  - End-to-end learning for the inference/mapping function
  - Deeper architecture for strong and distinct image priors
- Architecture design
  - Residual learning or other structures
  - Batch normalization and other network regularizations
  - Various blocks, e.g., Conv, Deconv, Pooling, …
- Optimization algorithms
  - SGD, momentum SGD, Adam
- Speed
  - GPU

General pipeline: training

- Training Phase

The problem →	***Architecture design*** →	***Prepare training data*** →	***Model training***
Denoising; super-resolution; debluring; …	Network structure; loss function; receptive filed; …	Degraded-latent sample pairs; degradation model parameters	Learn to predict the latent image with the given input

Input: degraded-latent sample pairs (and ***H***)
Output: Trained model

General pipeline: testing

- Testing Phase

Input: Degraded images
Output: Restored images

Super-resolution via CNN (SRCNN)

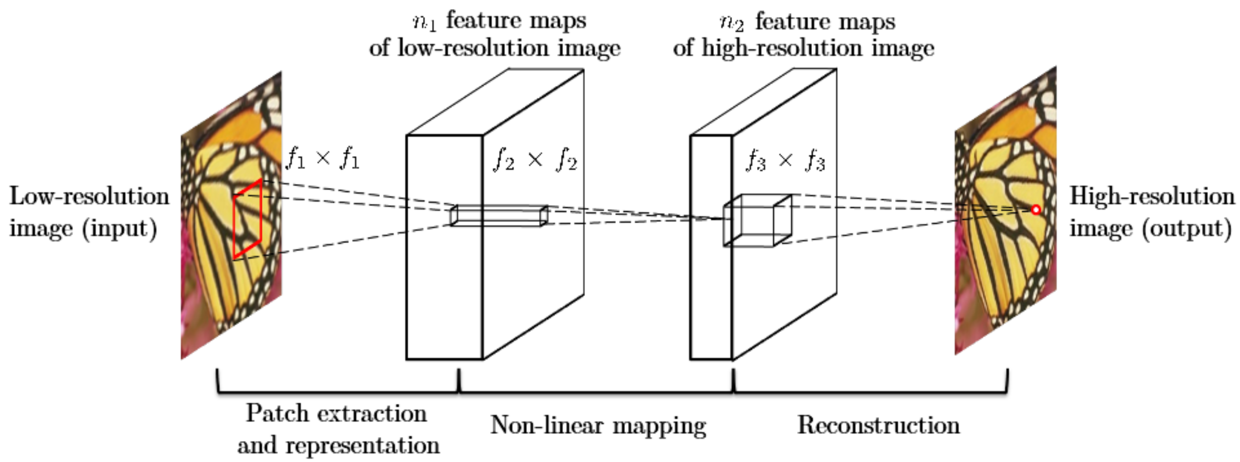
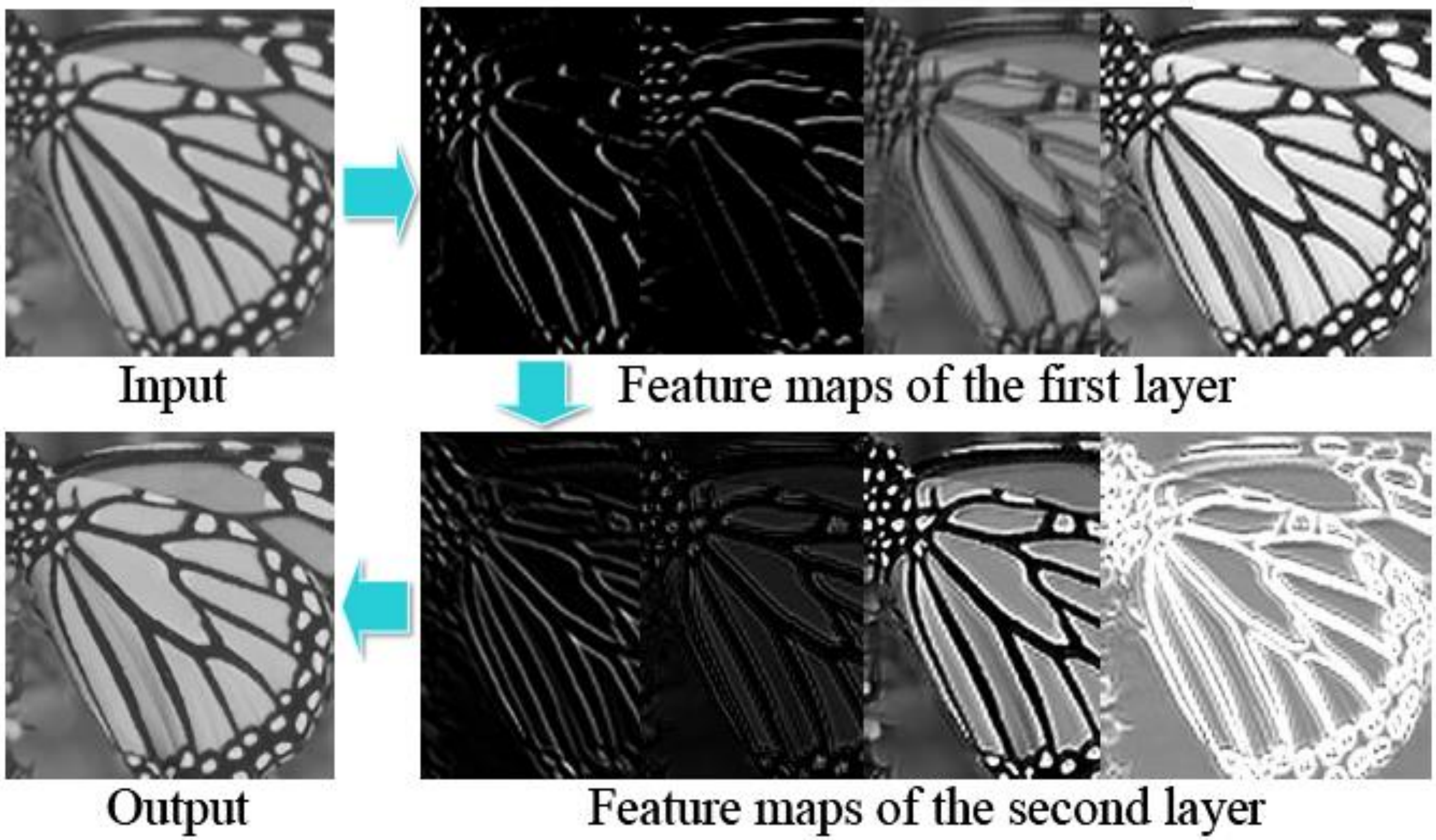


Fig. 2. Given a low-resolution image $\mathbf{Y}$, the first convolutional layer of the SRCNN extracts a set of feature maps. The second layer maps these feature maps nonlinearly to high-resolution patch representations. The last layer combines the predictions within a spatial neighbourhood to produce the final high-resolution image $F(\mathbf{Y})$.

256×256 (input, bicubic interpolation) → 256 × 256 × 64 (feature map of Conv1) → 256 × 256 × 32 (feature map of Conv2) → 256 × 256 (output)

Dong, Chao, et al. "Image super-resolution using deep convolutional networks." *IEEE PAMI* 38.2 (2016): 295-307.

SRCNN: example feature maps



Very deep CNN for SR (VDSR)

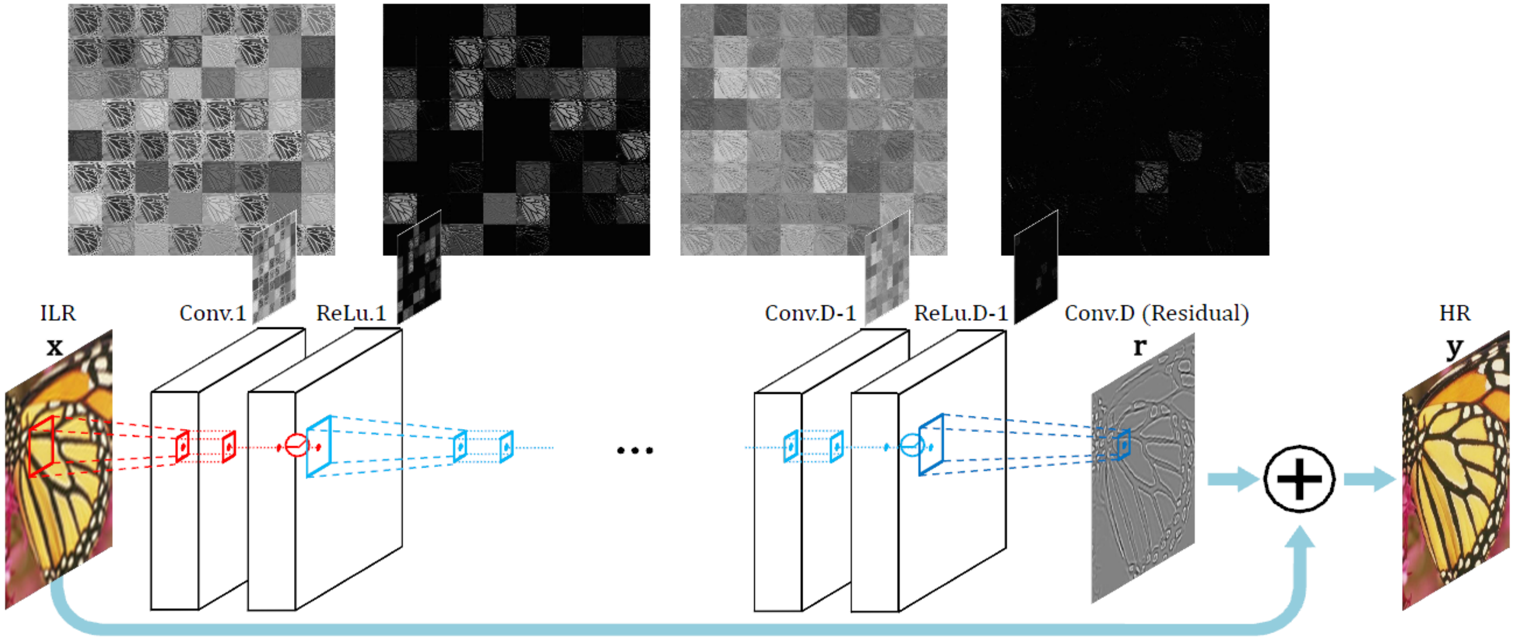


Figure 2: Our Network Structure. We cascade a pair of layers (convolutional and nonlinear) repeatedly. An interpolated low-resolution (ILR) image goes through layers and transforms into a high-resolution (HR) image. The network predicts a residual image and the addition of ILR and the residual gives the desired output. We use 64 filters for each convolutional layer and some sample feature maps are drawn for visualization. Most features after applying rectified linear units (ReLu) are zero.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks." *CVPR*, 2016.

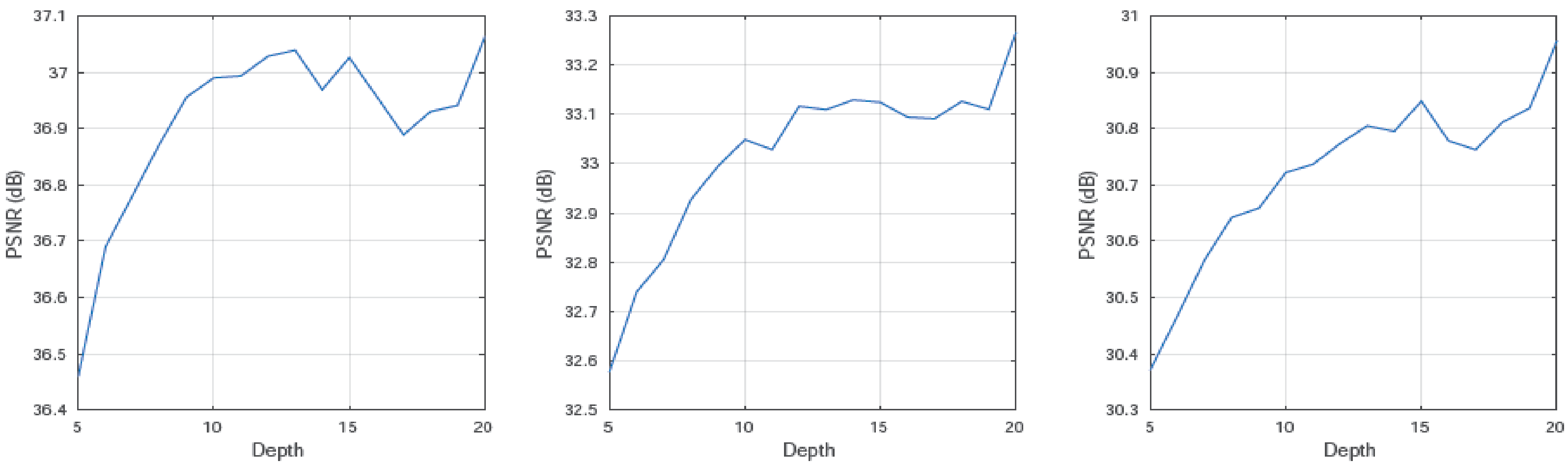
VDSR with and without residual learning

Epoch	10	20	40	80
Residual	36.74	36.87	36.91	36.93
Non-Residual	30.33	33.59	36.26	36.42
Difference	6.41	3.28	0.65	0.52

Performance table (PSNR) for residual and non-residual networks ('Set5' dataset, X2). Residual networks rapidly approach their convergence within 10 epochs.

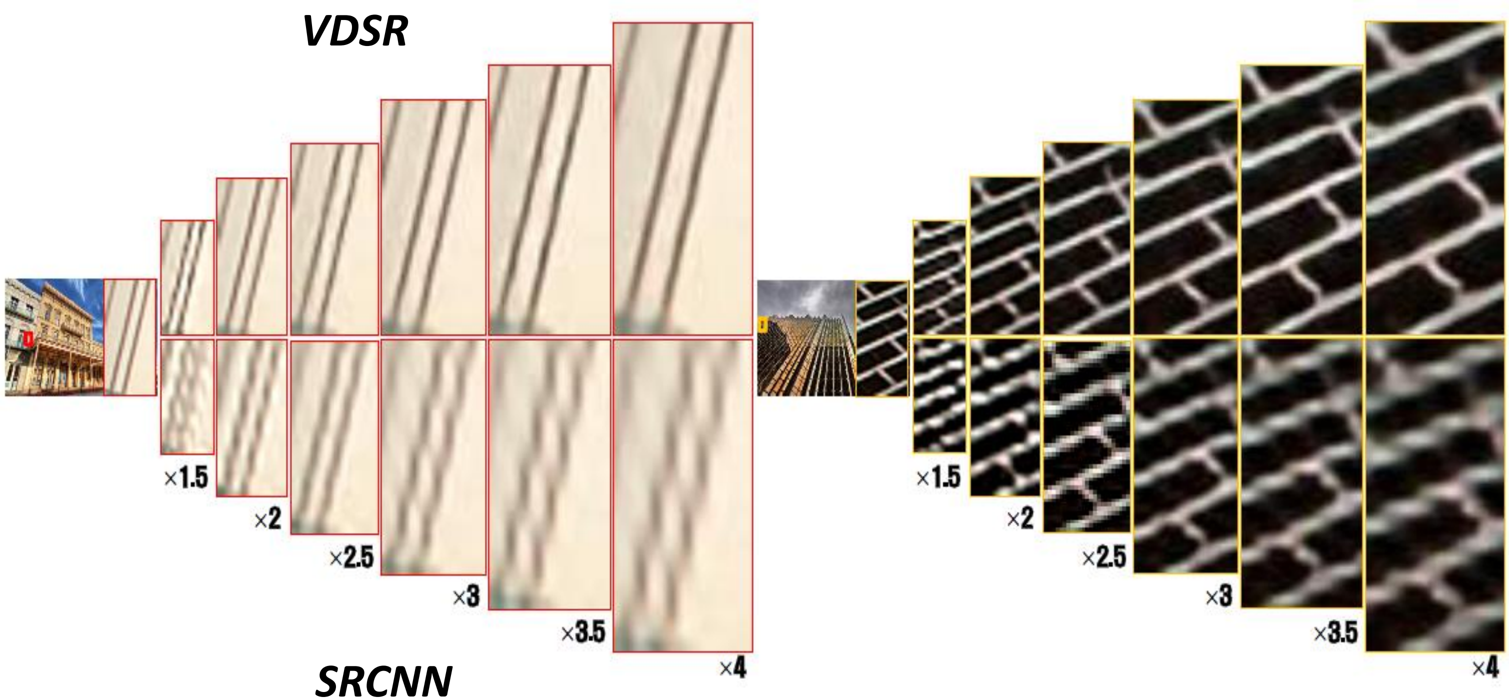
Main points of VDSR

- Residual learning is effective
- The deeper, the better
- Single network for multiple scaling factors

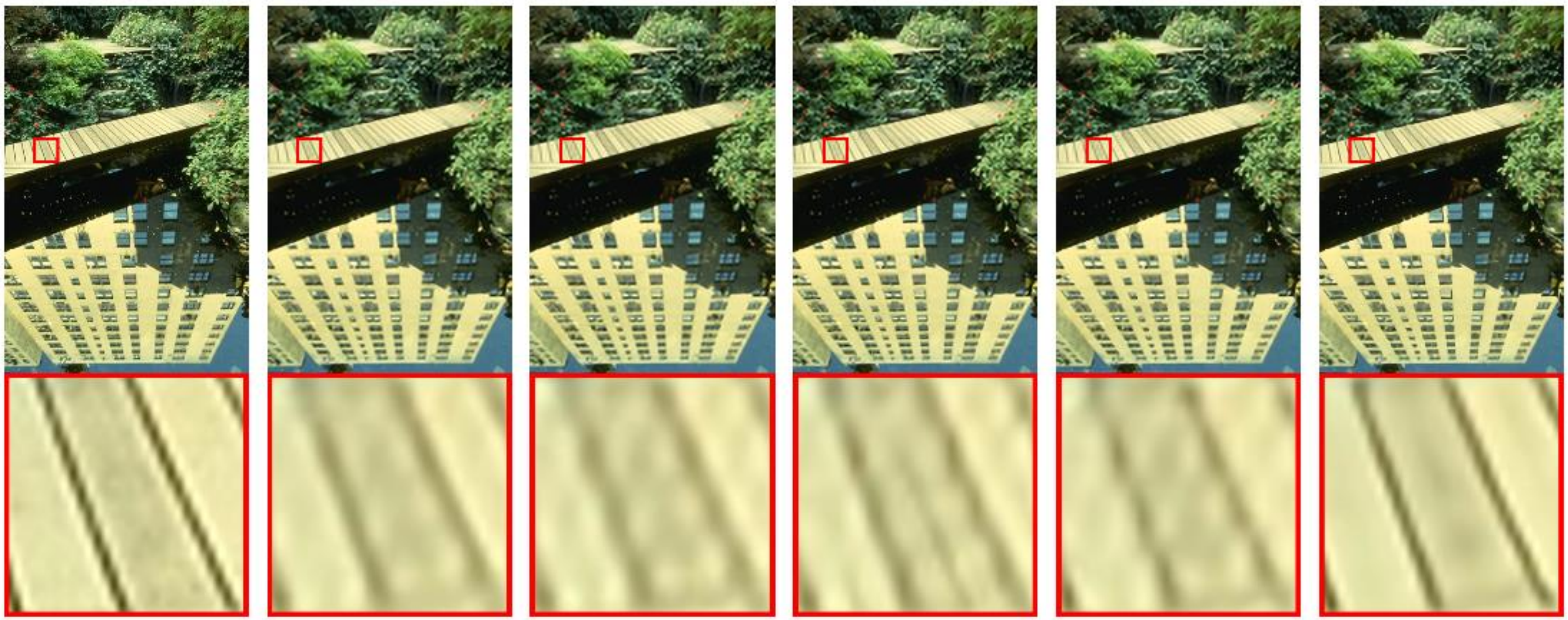


(a) Test Scale Factor 2 (b) Test Scale Factor 3 (c) Test Scale Factor 4

VDSR: single network, multiple scaling factors

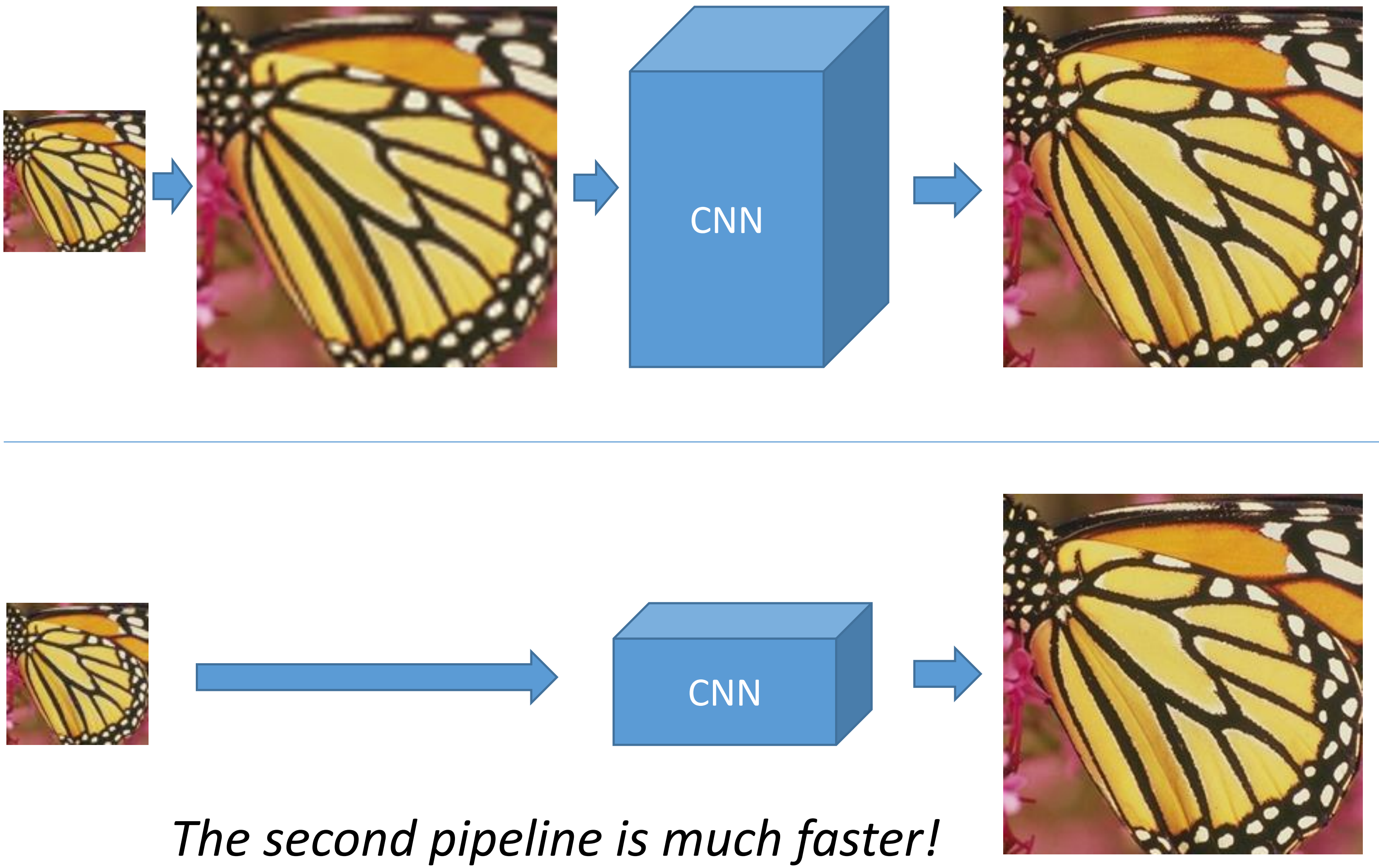


VDSR: examples



Ground Truth (PSNR, SSIM)	A+ [22] (22.92, 0.7379)	RFL [18] (22.90, 0.7332)	SelfEx [11] (23.00, 0.7439)	SRCNN [5] (23.15, 0.7487)	VDSR (Ours) (23.50, 0.7777)

Drawback of SRCNN and VDSR



The second pipeline is much faster!

Efficient sub-pixel CNN (ESPCNN)

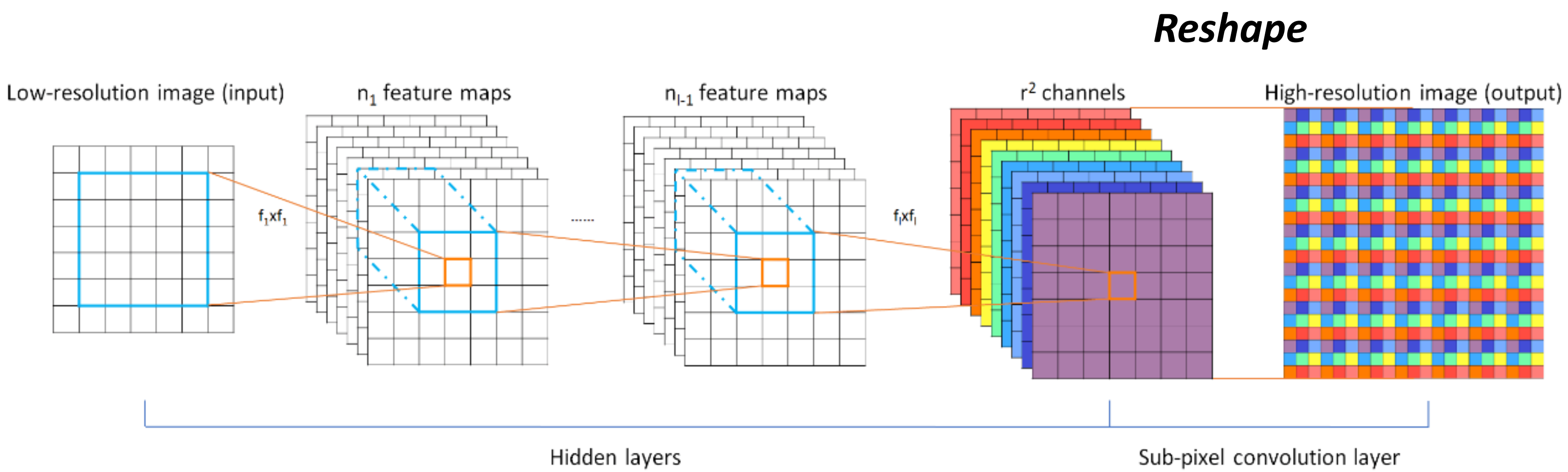
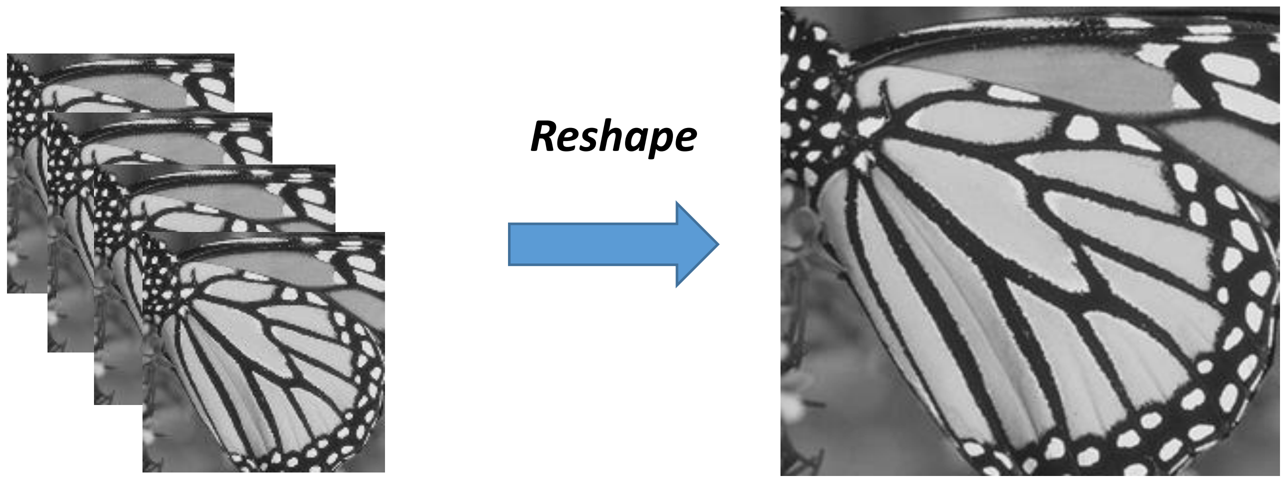


Figure 1. The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

Wenzhe Shi, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." *CVPR*, 2016.

ESPCNN: last layer



Last layer of ESPCN (X2)

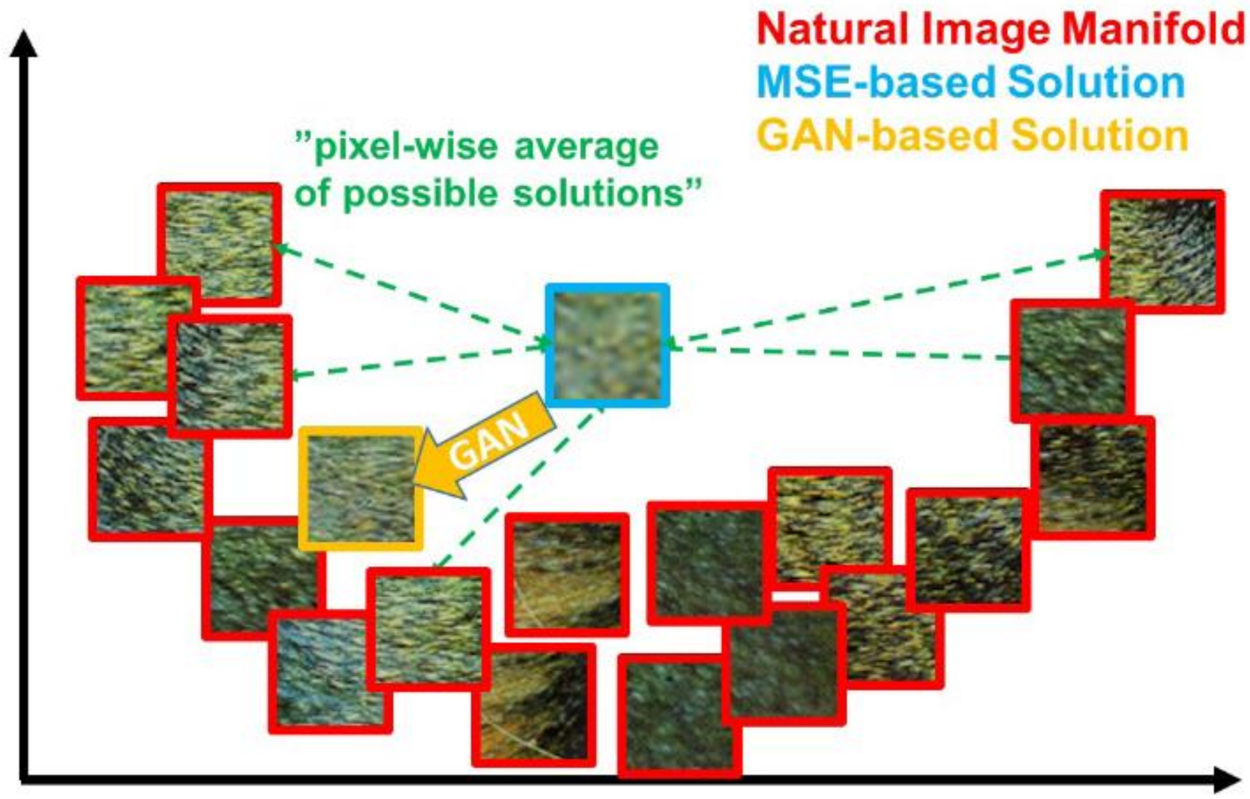
Is PSNR a good metric for SR?

State-of-the-art by PSNR

How about this one?

Scaling factor: x4

SR by GAN (SRGAN): motivation



- ***MSE-based solution** appears **overly smooth** due to the **pixel-wise average** of possible solutions in the pixel space.*
- ***Using GAN (Generative Adversarial Network) to drive the reconstruction towards the natural image manifold producing perceptually more convincing solutions.***

SRGAN

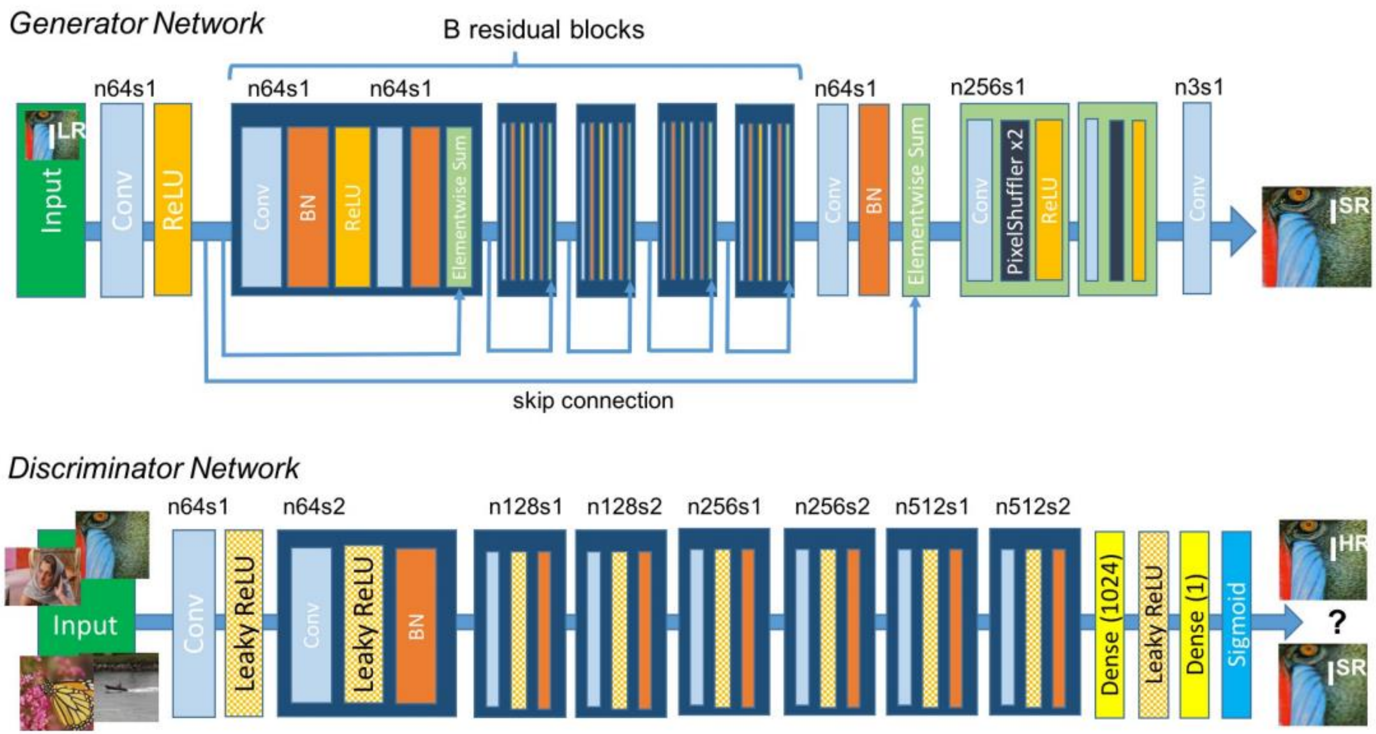


Figure 4: Architecture of Generator and Discriminator Network with corresponding number of feature maps (n) and stride (s) indicated for each convolutional layer.

C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", *CVPR*, 2017

SRGAN: perceptual loss function

Perceptual loss = content loss + adversarial loss

Content loss

$$l_{VGG/content}(\theta_G) = \frac{1}{2}\left\|\varphi\left(I^{HR}\right) - \varphi\left(Generator_{\theta_G}\left(I^{LR}\right)\right)\right\|_F^2$$

φ: feature map

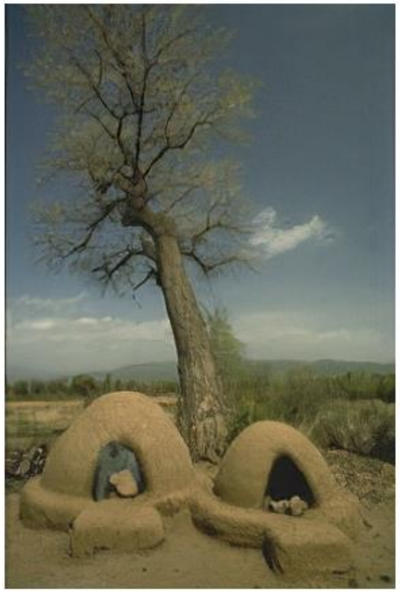
Adversarial loss

$$l_{Adaversarial}(\theta_G) = \sum_{n=1}^{N} -Discriminator_{\theta_D}\left(Generator_{\theta_G}\left(I_n^{LR}\right)\right)$$

SRGAN: examples

Bicubic

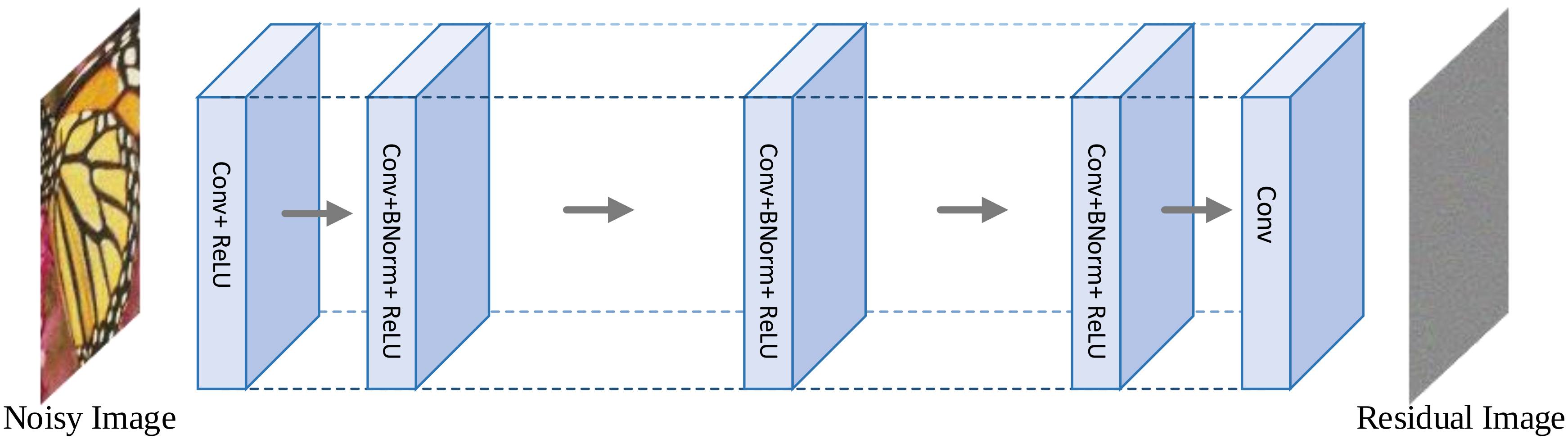
SRResNet



SRGAN

Ground-truth

DnCNN: deep residual learning beyond Gaussian denoising

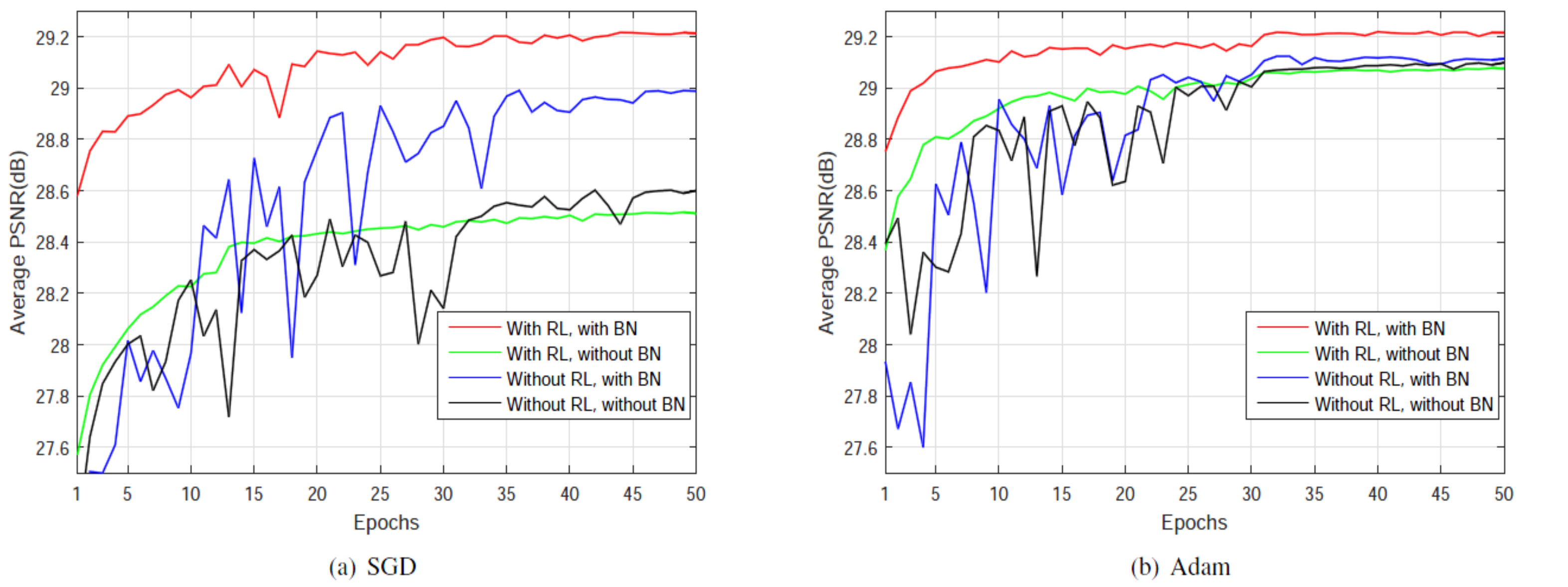


- Batch normalization and residual learning are particularly beneficial to Gaussian noise removal
- Single model for multiple tasks

K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Trans. on Image Processing*, 2017.

Code: https://github.com/cszn/DnCNN

Effect of batch normalization and residual learning



The Gaussian denoising results of four models under two gradient-based optimization algorithms, i.e., (a) SGD, (b) Adam, with respect to epochs. The four specific models are in different combinations of residual learning (RL) and batch normalization (BN) and are trained with noise level 25. The results are evaluated on 68 natural images from Berkeley segmentation dataset.

Gaussian denoising results

The averaged PSNR(dB) results of different methods on BSD68 dataset.

Methods	BM3D	WNNM	EPLL	MLP	CSF	TNRD	DnCNN
15	31.07	31.37	31.21	-	31.24	31.42	**31.73**
25	28.57	28.83	28.68	28.96	28.74	28.92	**29.23**
50	25.62	25.87	25.67	26.03	-	25.97	**26.23**

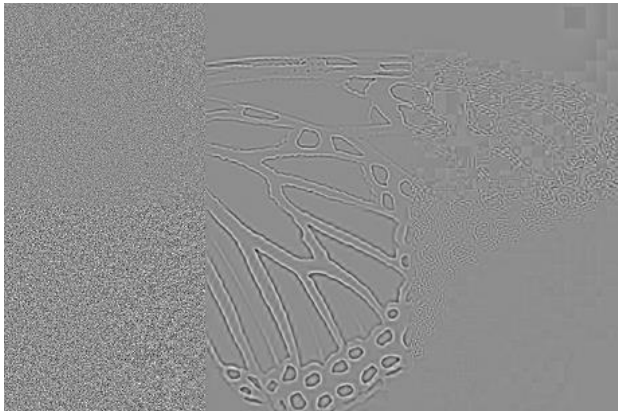
(a) Noisy / 14.76dB (b) BM3D / 26.21dB (c) WNNM / 26.51dB (d) TNRD / 26.59dB (e) DnCNN / 26.92dB

A single model for multiple tasks

Gaussian Denoising				
Dataset	**Noise Level**	**BM3D**	**TNRD**	**DnCNN-3**
BSD68	15	31.08	31.42	**31.46**
	25	28.57	28.92	**29.02**
	50	25.62	25.97	**26.10**
Single Image Super-Resolution				
Dataset	**Scale**	**TNRD**	**VDSR**	**DnCNN-3**
Set5	2	36.86	37.56	**37.58**
	3	33.18	33.67	**33.75**
	4	30.85	31.35	**31.40**
Set14	2	32.51	33.02	**37.58**
	3	29.43	29.77	**29.81**
	4	27.66	27.99	**28.04**
JPEG Image Deblocking				
Dataset	**Quality**	**ARCNN**	**TNRD**	**DnCNN-3**
LIVE1	10	28.96	29.28	**29.19**
	20	31.29	31.47	**31.59**
	30	32.67	32.78	**32.98**
	40	33.63	-	**33.96**

An example

Input image



Output residual image

Restored image

Gaussian denoising, single image super-resolution and JPEG image deblocking via **a single model**!

IterCNN for deblurring

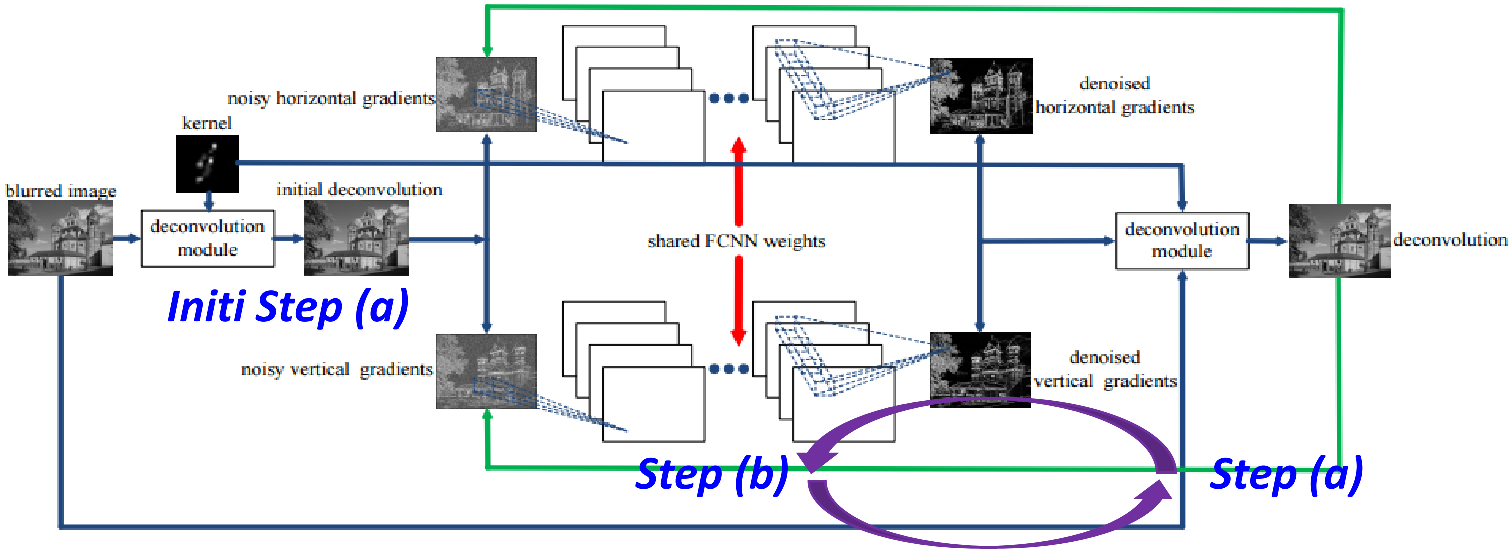


Figure 1. Network structure. Our network first deconvolves blurry input images by the deconvolution module and then performs convolutions to the vertical and horizontal gradients to generate the results with fewer noises. Finally, the deconvolution module is applied to the denoised gradients to generate the clear images. See text for more details.

$$min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{k} * \boldsymbol{x}\|_2^2 + \lambda \cdot \sum_{\boldsymbol{l}=\boldsymbol{h},\boldsymbol{w}} R(\boldsymbol{p_l} * \boldsymbol{x})$$

$\boldsymbol{p_h}$ and $\boldsymbol{p_w}$ are horizontal and vertical gradient operators.

Jiawei Zhang, et al. "Learning Fully Convolutional Networks for Iterative Non-blind Deconvolution." CVPR, 2017.

One motivation

- Model based optimization methods
  - ✓General to handle different image restoration problems
  - × The hand-crafted prior may not be strong enough
- Discriminative learning based methods
  - ✓Data driven end-to-end learning
  - × The generality of learned models is limitted
- Can we integrate the model based optimization and discriminative learning to develop a general image restoration method?

Half quadratic splitting

- The general model for image restoration

$$min_{\boldsymbol{x}}\ 0.5\|\boldsymbol{y}-\boldsymbol{Hx}\|_2^2\ +\lambda\cdot R(\boldsymbol{x})$$

- Introducing an auxiliary variable $\boldsymbol{z}$ ($\boldsymbol{z}\approx\boldsymbol{x}$)

$$min_{\boldsymbol{x},\boldsymbol{z}}\ 0.5\|\boldsymbol{y}-\boldsymbol{Hx}\|_2^2\ +\lambda\cdot R(\boldsymbol{z})+0.5\mu\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

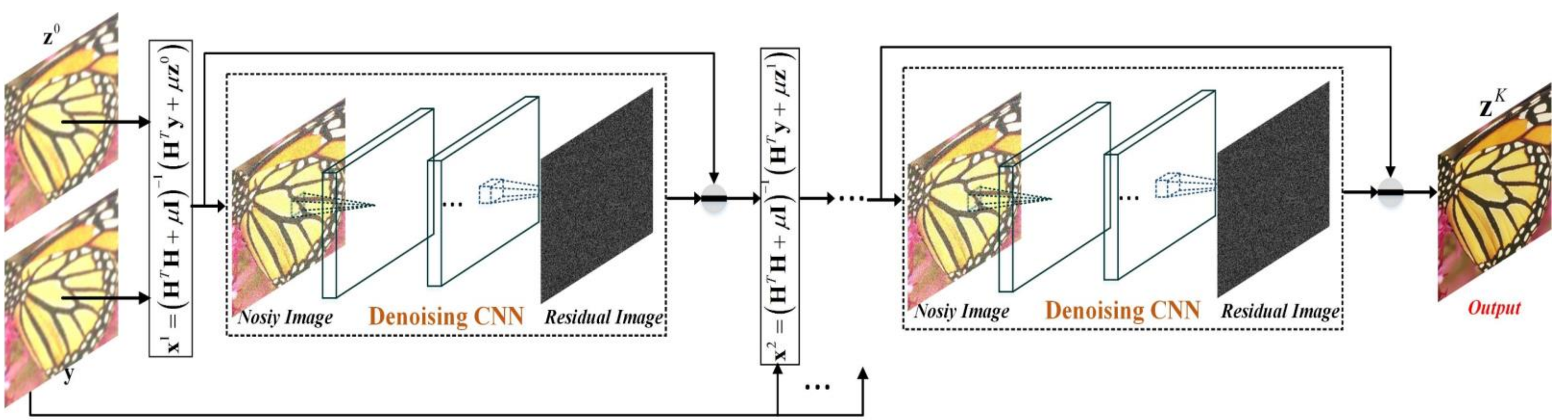
- Solving $\boldsymbol{x}$ and $\boldsymbol{z}$ alternatively and iteratively

(a) $min_{\boldsymbol{x}}\ \|\boldsymbol{y}-\boldsymbol{Hx}\|_2^2\ +\mu\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$ % Data proximal operator

(b) $min_{\boldsymbol{z}}\ 0.5\mu\|\boldsymbol{x}-\boldsymbol{z}\|_2^2+\lambda\cdot R(\boldsymbol{z})$ % Denoising sub-problem

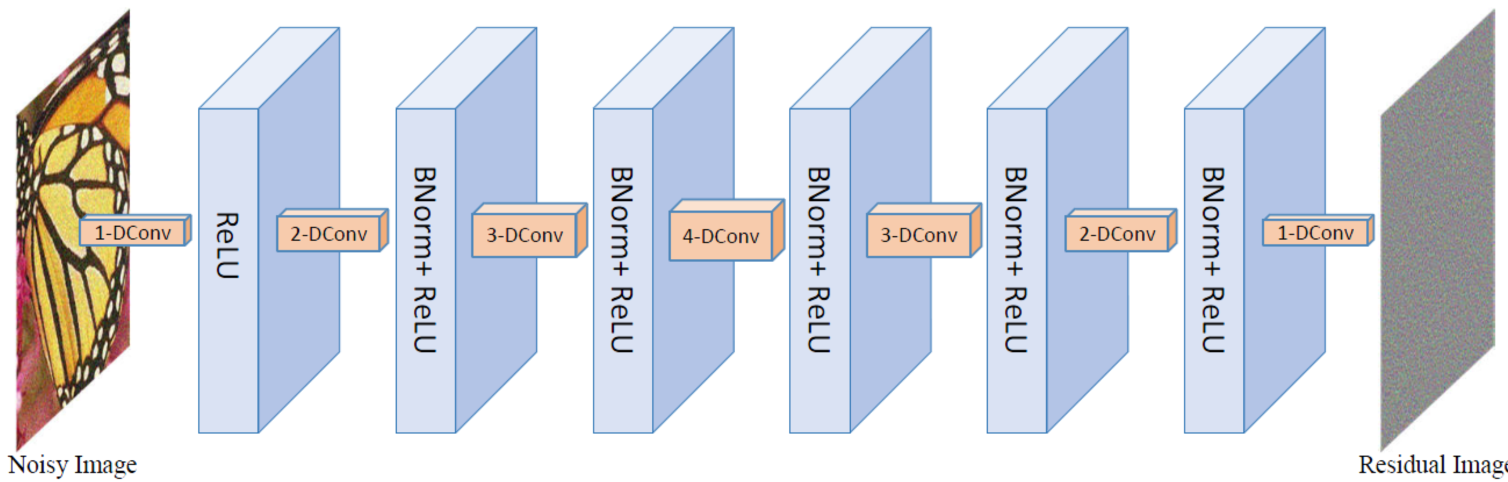
Image restoration with deep CNN denoiser prior (IRCNN)

- Plugging the strong CNN denoiser prior into model-based optimization
- Step (a): analytical solution
- Step (b): deep CNN denoiser



K. Zhang, W. Zuo, S. Gu, L. Zhang. "Learning Deep CNN Denoiser Prior for Image Restoration." *CVPR* 2017.
Code: https://github.com/cszn/ircnn

CNN denoiser



"*s*-DConv" denotes *s*-dilated convolution, $s = 1, 2, 3$ and 4. A dilated filter with dilation factor s can be simply interpreted as a sparse filter of size $(2s+1) \times (2s+1)$ where only 9 entries of fixed positions are non-zeros.

Denoising results

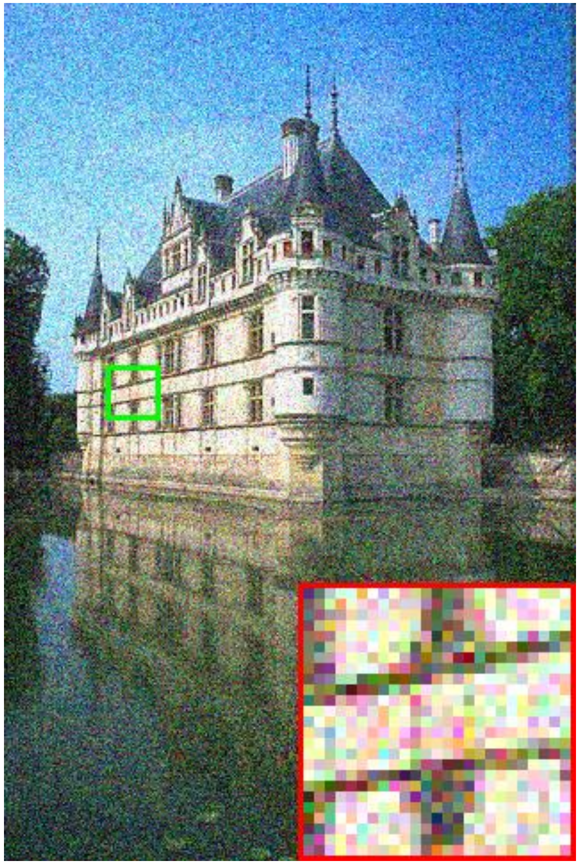
The average PSNR(dB) results of different methods on (gray) BSD68 dataset.

Methods	BM3D	WNNM	TNRD	MLP	IRCNN
σ=15	31.07	31.37	31.42	-	31.63
σ=25	28.57	28.83	28.92	28.96	29.15
σ=50	25.62	25.87	25.97	26.03	26.19

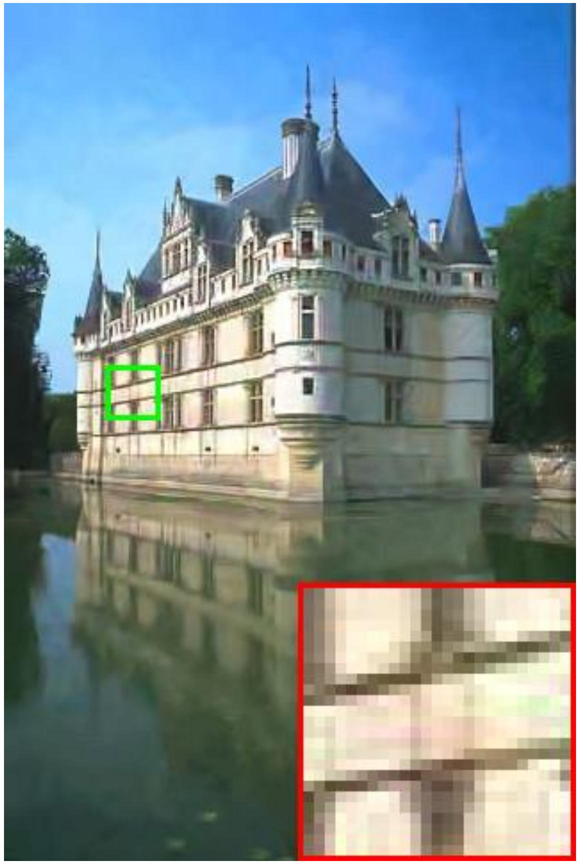
The average PSNR(dB) results of CBM3D and proposed CNN denoiser on (color) BSD68 dataset.

Noise Level	5	15	25	35	50
CBM3D	40.24	33.52	30.71	28.89	27.38
IRCNN	40.36	33.86	31.16	29.50	27.86

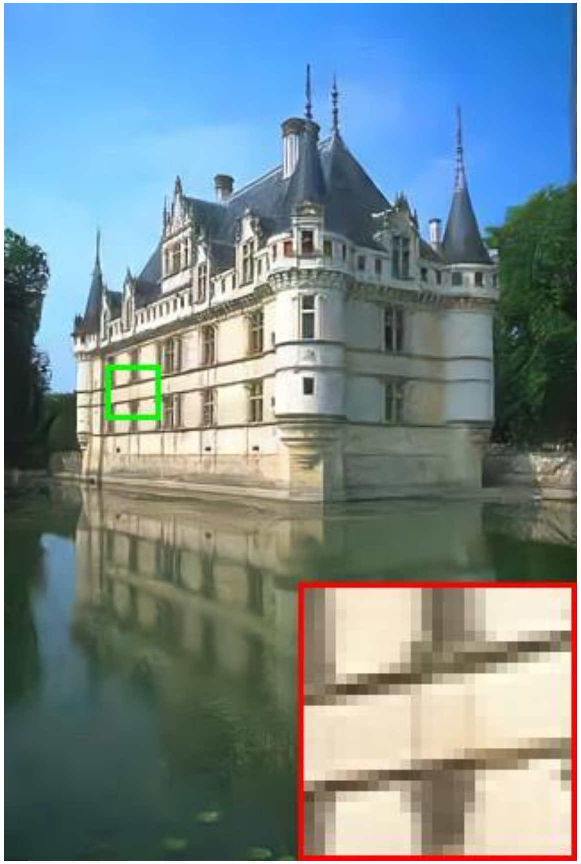
Denoising results



Noisy (17.75dB)



CBM3D (29.90dB)



IRCNN (30.42dB)

Denoising results

Noisy (17.70dB)

CBM3D (27.25dB)

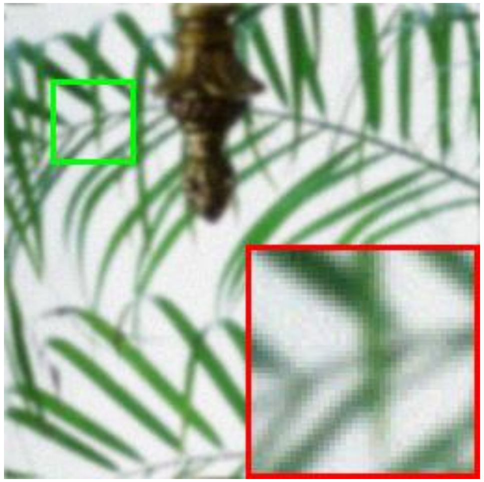
IRCNN (28.06dB)

Debluring results

Methods	σ	*C.man*	*House*	*Lena*	*Monar.*	*Leaves*	*Parrots*
Gaussian blur with standard deviation 1.6							
IDDBM3D	2	27.08	32.41	30.28	27.02	26.95	30.15
NCSR		27.99	33.38	30.99	28.32	27.50	30.42
MLP		27.84	33.43	31.10	28.87	28.91	31.24
IRCNN		28.12	33.80	31.17	30.00	29.78	32.07
Kernel 1 (19×19)							
EPLL	2.55	29.43	31.48	31.68	28.75	27.34	30.89
IRCNN		32.07	35.17	33.88	33.62	33.92	35.49
EPLL	7.65	25.33	28.19	27.37	22.67	21.67	26.08
IRCNN		28.11	32.03	29.51	29.20	29.07	31.63
Kernel 2 (17×17)							
EPLL	2.55	29.67	32.26	31.00	27.53	26.75	30.44
IRCNN		31.69	35.04	33.53	33.13	33.51	35.17
EPLL	7.65	24.85	28.08	27.03	21.60	21.09	25.77
IRCNN		27.70	31.94	29.27	28.73	28.63	31.35

Debluring results

Ground-truth



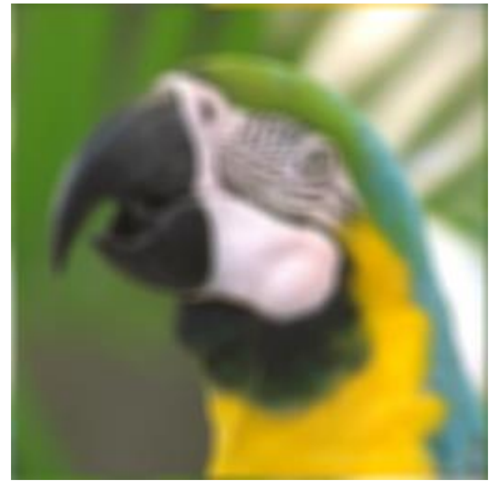
Blurred and noisy

IDDBM3D (25.32dB)

IRCNN (27.89dB)

Debluring results

Ground-truth



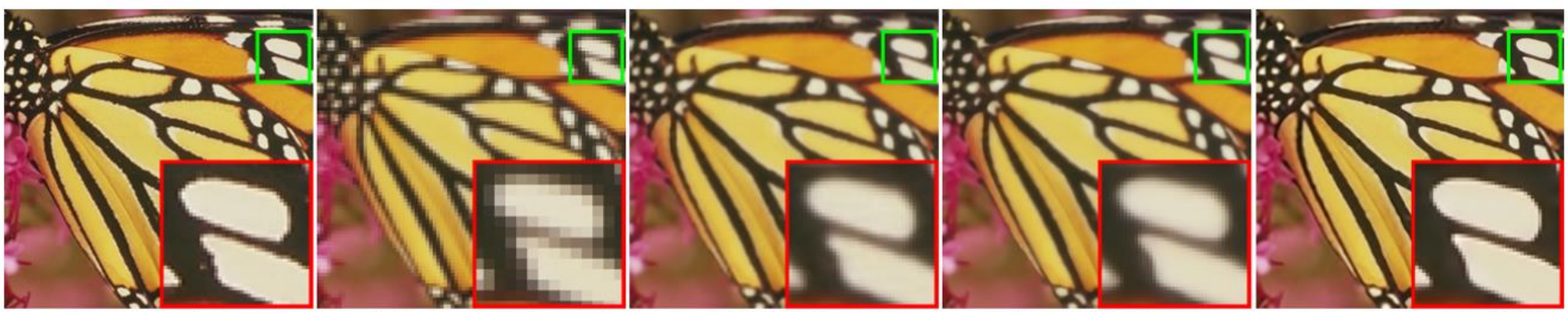
Blurred and noisy

NCSR (29.00dB)

IRCNN (31.65dB)

Super-resolution results

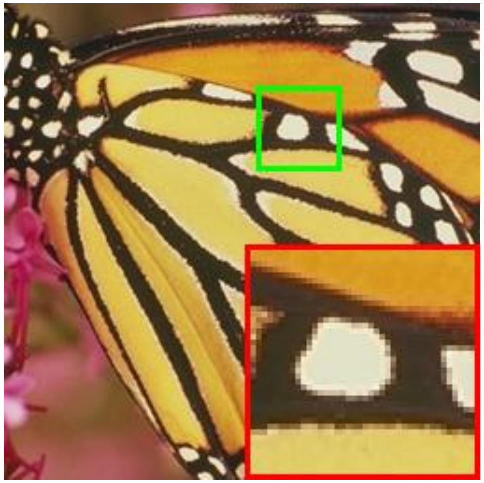
Dataset	Scale	Kernel	Channel	SRCNN	VDSR	NCSR	SPMSR	SRBM3D	$SRBM3D_G$	$SRBM3D_C$	$Proposed_G$	$Proposed_C$
Set5	2	Bicubic	Y	36.65	37.56	-	36.11	37.10	36.34	36.25	37.40	37.26
			RGB	34.45	35.16	-	33.94	-	34.11	34.22	35.02	35.12
	3	Bicubic	Y	32.75	33.67	-	32.31	33.30	32.62	32.54	33.35	33.22
			RGB	30.72	31.50	-	30.32	-	30.57	30.69	31.23	31.30
	3	Gaussian	Y	30.42	30.54	33.02	32.27	-	32.66	32.59	33.39	33.26
			RGB	28.50	28.62	30.00	30.02	-	30.31	30.74	30.93	31.35
Set14	2	Bicubic	Y	32.43	33.02	-	31.96	32.80	32.09	32.25	32.85	32.85
			RGB	30.43	30.90	-	30.05	-	30.15	30.32	30.76	30.84
	3	Bicubic	Y	29.27	29.77	-	28.93	29.60	29.11	29.27	29.58	29.55
			RGB	27.44	27.85	-	27.17	-	27.32	27.47	27.70	27.72
	3	Gaussian	Y	27.71	27.80	29.26	28.89	-	29.18	29.39	29.63	29.62
			RGB	26.02	26.11	26.98	27.01	-	27.24	27.60	27.59	27.80



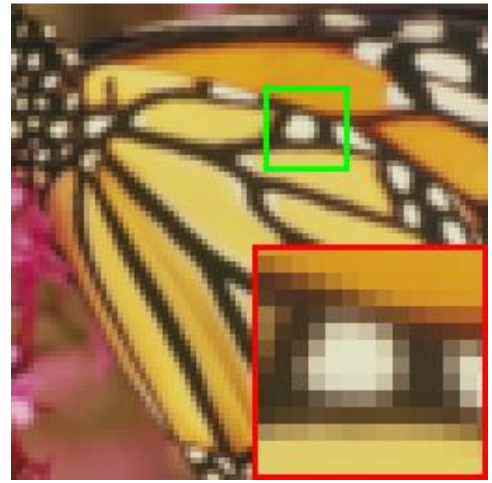
(a) Ground-truth (b) Zoomed LR image (c) SRCNN (24.46dB) (d) VDSR (24.73dB) (e) $Proposed_G$ (29.30dB)

Single image super-resolution performance comparison for *Butterfly* image from Set5 (the blur kernel is 7×7 Gaussian kernel with standard deviation 1.6, the scale factor is 3).

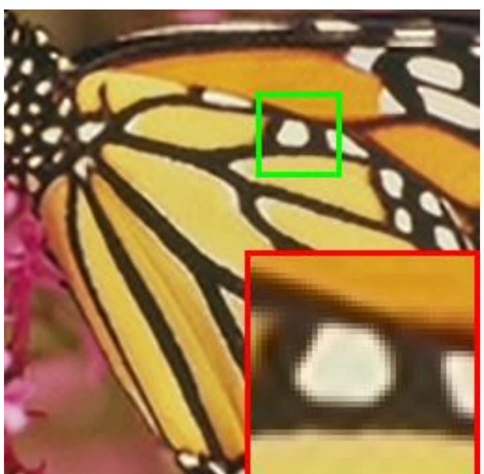
Super-resolution results



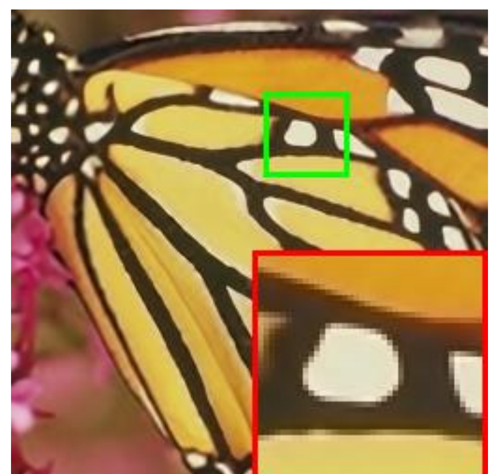
Ground-truth



Zoomed LR image

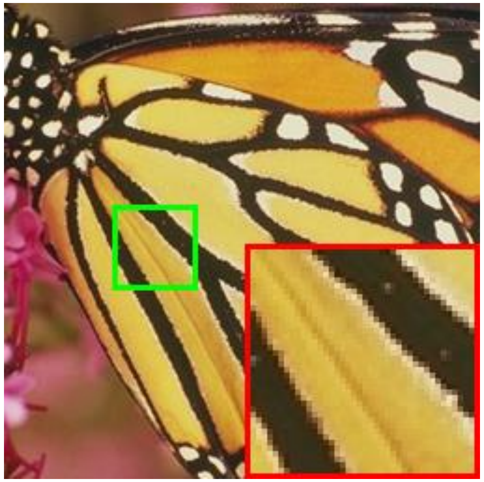


NCSR (28.05dB)

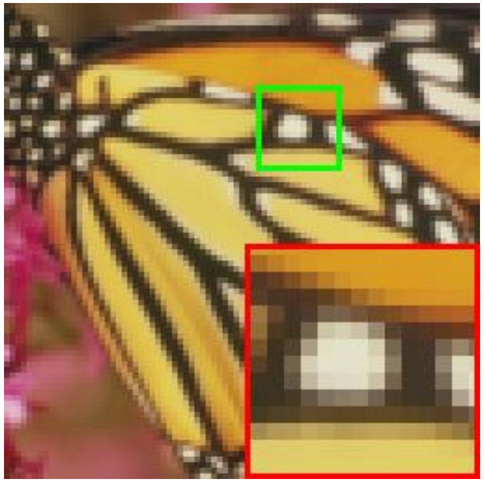


IRCNN (29.32dB)

Super-resolution results

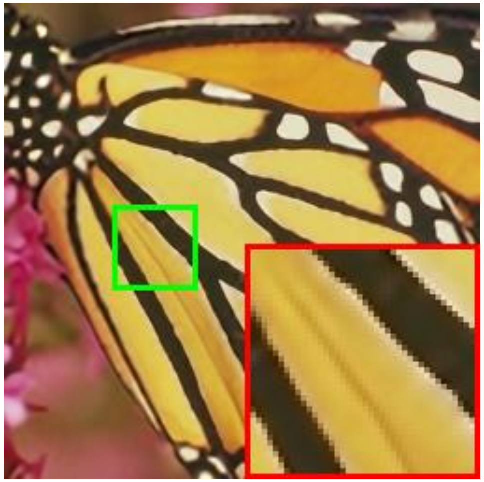


Ground-truth



Zoomed LR image

BM3D (26.88dB)

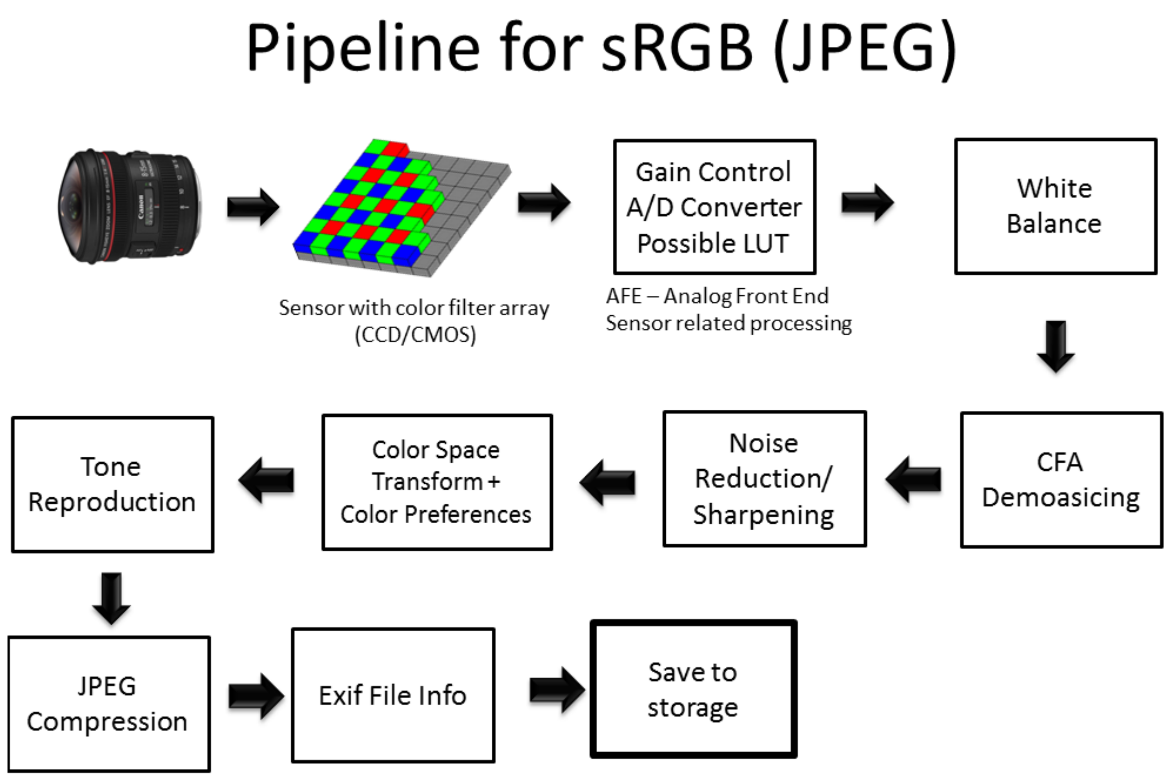


IRCNN (29.29dB)

Open problems

Camera pipeline

- The digital imaging process is very complex

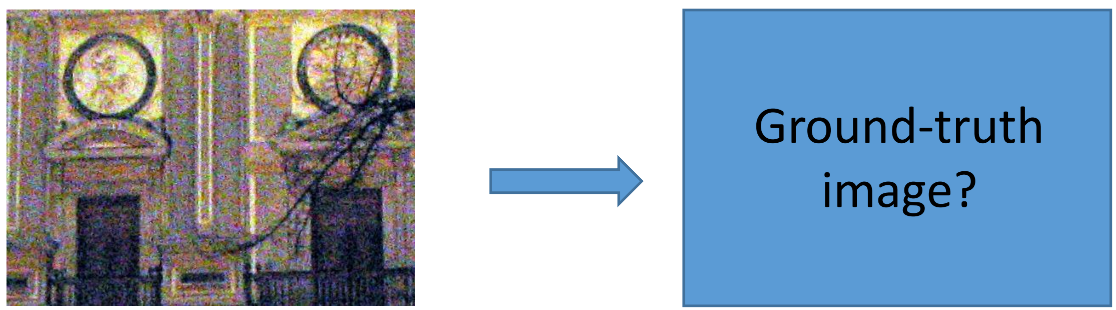


Blind real image restoration

- The degradations in real images are too complex to be described by simple models
  - Non-Gaussian noise, signal dependent, non-uniform blur, compression artifacts, system distortions, …

Deep learning?

- Deep learning for blind real image restoration!?
- Good idea! But where are the ground-truth images for supervised learning?



- How can we do deep learning based image restoration without paired data?
- Is GAN a solution for this challenging problem?

Summary

Summary

- Image sparsity and low-rankness priors have been dominantly used in past decades.
- Recently the CNN based models have been rapidly developed to learn deep image priors.
- There remain many challenging issues for deep learning based image restoration.
  - ➢ Key issue: the lack of training image pairs in real-world blind image restoration applications.
- It is still an open problem to train deep image restoration models without using image pairs.

THANKS

PolyU

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Department of Computing
電子計算學系