Deep Super-Resolution Network for Single Image Super-Resolution with Realistic Degradations

Rao Muhammad Umer, Gian Luca Foresti, Christian Micheloni

University of Udine

University of Udine – Uniud
Machine Learning and Perception Lab – MLP
Artificial Vision and Real-Time Systems Lab – AViReS

13th International Conference on Distributed Smart Cameras – ICDSC

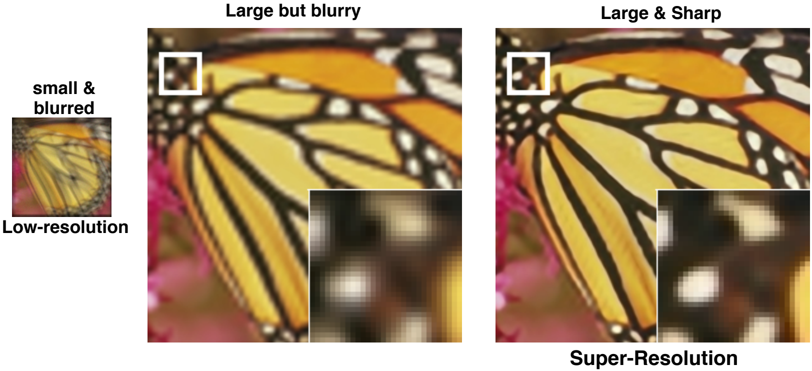
September 10, 2019

Outlines

1. Introduction to Super-Resolution Problem
2. Related Works
3. Our Approach
   - Problem Formulation
   - Optimization Strategy
4. Proposed Network
   - Deconvolution module
   - Denoising module
   - Upscaling module
5. Training details
6. Experimental Results
   - Quantitative Results
   - Computational Performance
   - Visual Results
7. Conclusion

Introduction to Super-Resolution Problem I

- Single Image Super-Resolution (SISR) Problem:



Introduction to Super-Resolution Problem II

- Bicubic degradation model:

$$\mathbf{y} = \mathbf{x} \downarrow_s, \tag{1}$$

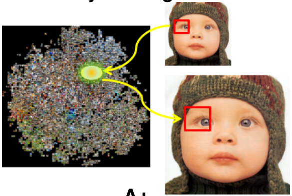
- General degradation model:

$$\mathbf{y} = (\mathbf{k} * \mathbf{x}) \downarrow_s + \mathbf{n}, \tag{2}$$

- The goal is to enlarge an image with details recovered.
- Highly ill-posed inverse problem (many possible solutions) due to unknown noise and loss of high-frequency information (*i.e.* edges, texture).

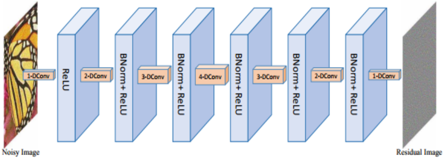
Related Works

Dictionary-learning based method



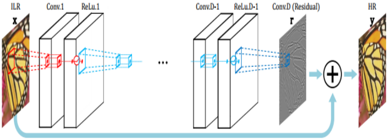
A+

Deep Learning based method



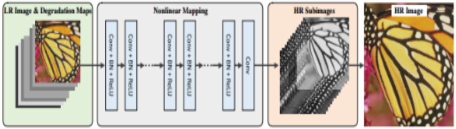
IRCNN

Deep Learning based method



VDSR

Deep Learning based method



SRMD

Our Approach I

- Problem Formulation:
  - More realistic degradation model:

$$\mathbf{y} = \mathbf{k} * (\mathbf{x} \downarrow_s) + \mathbf{n}, \tag{3}$$

  - Formulate the energy function according to Maximum A Posteriori (MAP) framework as:

$$\hat{\mathbf{x}} = \arg\min_x \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{k} * (\mathbf{x} \downarrow_s)\|_2^2 + \lambda\varphi(\mathbf{x}), \tag{4}$$

Our Approach II

- Optimization Strategy:
  - We want to recover the underlying image $\mathbf{x}$ as the minimizer of the objective function as:

$$\hat{\mathbf{x}} = \arg\min_{x} \mathbf{E}(\mathbf{x}), \tag{5}$$

$$\hat{\mathbf{x}} = \arg\min_{x} \mathbf{D}(\mathbf{x}; \mathbf{k}, \mathbf{y}, \downarrow_{\mathbf{s}}) + \lambda\varphi(\mathbf{x}), \tag{6}$$

$$\hat{\mathbf{x}} = \underbrace{\arg\min_{a \leqslant x \leqslant b} \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{k} * (\mathbf{x} \downarrow_s)\|_2^2 + \lambda\varphi(\mathbf{x})}_{\mathbf{f}(\mathbf{x})}, \tag{7}$$

$$\hat{\mathbf{x}} = \arg\min_{x} \mathbf{f}(\mathbf{x}) + \mathbf{i}_c(\mathbf{x}), \tag{8}$$

where $\mathbf{i}_c$ is the indicator function of the convex set $\mathbf{C} \in \{\mathbf{x} \in \mathbb{R}^m : \mathbf{a} \leqslant \mathbf{x}_k \leqslant \mathbf{b}, \forall k\}$.

Our Approach III

- The gradient of $\mathbf{f}(\mathbf{x})$ is computed in matrix-vector form as:

$$\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}) = \frac{1}{\sigma^2}\mathbf{K}^T(\mathbf{K}(\mathbf{x}\downarrow_s) - \mathbf{y}) + \lambda\Psi(\mathbf{x}), \tag{9}$$

- Proximal updates:

$$\mathbf{x}_t \downarrow_s = \mathrm{Prox}_{\gamma^t \mathbf{i}_c}\left(\mathbf{x}_{(t-1)\downarrow_s} - \gamma^t \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}_{(t-1)})\right), \tag{10}$$

where γ^t is a step-size and $\mathrm{Prox}_{\gamma^t \mathbf{i}_c}$ is the proximal operator [1] related to the indicator function i_c, which can be defined as:

$$\mathrm{Prox}_h(\mathbf{z}) = \arg\min_{x\in\mathbf{C}} \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|_2^2 + h(\mathbf{x}), \tag{11}$$

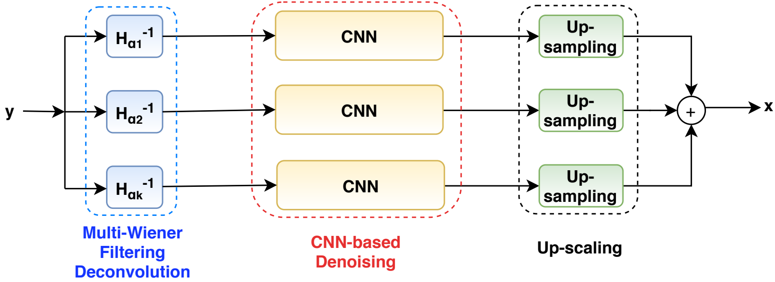
- Since proximal map $\mathrm{Prox}_{\gamma\sigma^2}$ gives the regularized solution of a Gaussian denoising problem, so finally we have the following form of our solution as:

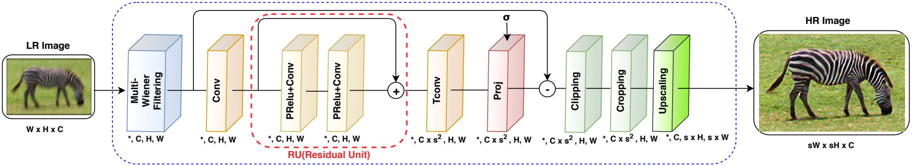
$$\mathbf{x}_t = \left(\mathrm{Prox}_{\gamma^t\sigma^2}\left((1-\gamma^t\mathbf{K}^T\mathbf{K})(\mathbf{x}_{(t-1)})\downarrow_s + \gamma^t\mathbf{K}^T\mathbf{y} - \lambda\gamma^t\Psi(\mathbf{x}_{t-1})\right)\right)\uparrow_s, \tag{12}$$

Our Approach IV

- The objective function is minimized by discriminative learning as:

$$\begin{cases} \arg\min_{\Theta} \mathcal{L}(\Theta) = \sum_{s=1}^{S} \frac{1}{2} \|\hat{\mathbf{x}}_T^s - \mathbf{x}_{gt}^s\|_2^2 \\ \text{s.t.} \begin{cases} \mathbf{x}_0^s = \mathbf{I}_0^s \\ update\ \mathbf{x}_t^s\ according\ to\ Eq.\ (12), \\ t = 1 \ldots T \end{cases} \end{cases} \tag{13}$$





- Deconvolution module:
  - **Multi-Wiener Filtering layer**: 24 output features map with kernel size 5×5 by initializing the discrete cosine transform (DCT) basis.
- Denoising module:
  - Motivated by UDNet [2] as a residual CNN denoiser.
  - **Residual Unit (RU) blocks**: used five blocks, which are sandwich by convolution (64 × 7 × 7) and transpose convolution (64 × 7 × 7) layer with shared parameters.

Proposed Network II

- **Reflection padding**: used before the *Wiener* and *Conv* layers to ensure slowly-varying changes at the boundaries of input images.
- **Projection layer** [2]: computes the proximal map for the indicator function (*i.e.* non-smooth part).
- **Clipping layer**: incorporates our prior knowledge about the valid range of image intensities and enforces the pixel values of the reconstructed image to lie in the range $[0, 255]$.
- **Cropping layer**: crops the spatial dimensions of the input image that is padded with the kernel dimension.

- Upscaling module:
  - Used **Sub-pixel convolution** [3] layer for upscaling features map.

Training details I

- **Training dataset**:
  - $\{\mathbf{x}_i, \mathbf{k}_i, \mathbf{y}_i\}_{i=1}^{N}$ by center cropped image patches with a size of 256×256 pixels from BSDS500 [4].
  - Bicubicly downsampling factors $\mathbf{s}$ (*i.e.* $\times 2, \times 3, \times 4$), motion blur kernels $\mathbf{k}$ with sizes range 11×11 to 31×31, Gaussian noises with 1% to 5% noise standard deviation to generate LR image patches.
- **Testing datasets**: Set5 [5], Set14 [5], and Urban100 [6].
- **ADAM optimizer setting**: lr=$1e^{-3}$, betas=(0.9, 0.999), eps=$1e^{-4}$, amsgrad=True
- **Loss function**:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_{grad}, \tag{14}$$

$$\mathcal{L}_{\mathrm{c}}(\mathbf{x}_i, \hat{\mathbf{x}}_i; \Theta) = \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2, \tag{15}$$

$$\mathcal{L}_{\mathrm{grad}}(\mathbf{x}_i, \hat{\mathbf{x}}_i; \Theta) = \|\nabla_v \hat{\mathbf{x}}_i - \nabla_v \mathbf{x}_i\|_2^2 + \|\nabla_h \hat{\mathbf{x}}_i - \nabla_h \mathbf{x}_i\|_2^2, \tag{16}$$

Training details II

- **Weights initialization**: He normal initialization [7] method to set the weights of the convolutional kernels and Wiener-layer kernel weights by DCT basis.
- Optimize the hyper-parameters and the weights of SRWDNet iteratively by avoiding local-minima to train the network in an end-to-end manner.

Experimental Results I

- Quantitative Results:

Dataset	Degradation Settings				Bicubic	VDSR [8] (CVPR-2016)	TNRD [9] (TPAMI-2017)	IRCNN [10] (CVPR-2017)	SRMD [11] (CVPR-2018)	SRWDNet (Ours)
	Scale Factor	Kernel size	Down-sampler	Noise Level	Average PSNR / SSIM					
Set5	$\times 2$	11×11 to 31×31	Bicubic	1%	19.30 / 0.5070	19.24 / 0.4767	19.41 / 0.4937	19.00 / 0.4545	17.94 / 0.4414	**23.13 / 0.5870**
	$\times 3$	11×11 to 31×31	Bicubic	1%	17.90 / 0.4668	17.86 / 0.4431	17.90 / 0.4765	17.63 / 0.4171	17.40 / 0.4311	**21.00 / 0.5025**
	$\times 4$	11×11 to 31×31	Bicubic	1%	17.01 / 0.4496	16.97 / 0.4296	17.21 / 0.4609	16.74 / 0.4053	16.72 / 0.4263	**20.58 / 0.5036**
Set14	$\times 2$	11×11 to 31×31	Bicubic	1%	18.85 / 0.4419	18.80 / 0.4147	18.99 / 0.4453	18.59 / 0.3981	17.15 / 0.3772	**21.28 / 0.5120**
	$\times 3$	11×11 to 31×31	Bicubic	1%	17.74 / 0.4127	17.70 / 0.3900	17.52 / **0.4726**	17.49 / 0.3722	17.24 / 0.3858	**19.25** / 0.4042
	$\times 4$	11×11 to 31×31	Bicubic	1%	16.99 / 0.4012	16.97 / 0.3818	17.10 / **0.4509**	16.75 / 0.3651	16.73 / 0.3842	**19.10** / 0.4109
Urban100	$\times 2$	11×11 to 31×31	Bicubic	1%	17.30 / 0.4007	17.25 / 0.3729	17.58 / 0.4336	17.01 / 0.4235	15.23 / 0.3357	**19.81 / 0.4914**
	$\times 3$	11×11 to 31×31	Bicublic	1%	16.44 / 0.3773	16.41 / 0.3539	16.45 / 0.4802	16.14 / 0.3523	15.85 / 0.3538	**17.98 / 0.3810**
	$\times 4$	11×11 to 31×31	Bicubic	1%	15.89 / 0.3694	15.87 / 0.3491	16.23 / **0.4608**	15.95 / 0.3478	15.65 / 0.3601	**17.65** / 0.3744

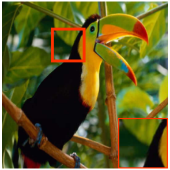
Experimental Results II

- Computational Performance [s]:

Degradation Scenario	VDSR [8] (CVPR-2016)	TNRD [9] (TPAMI-2017)	IRCNN [10] (CVPR-2017)	SRMD [11] (CVPR-2018)	SRWDNet (Ours)
image size: 500×480, motion blur kernel: 31×31, $\sigma = 1\%$, upscaling factor $= \times 4$	1.573	19.573	30.561	0.305	0.593

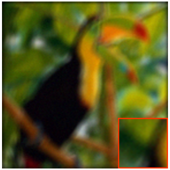
Experimental Results III

- Visual Results: $\times 2$ on Set5

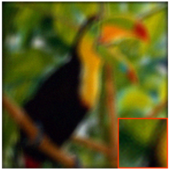


PSNR/SSIM
(a) Ground-truth

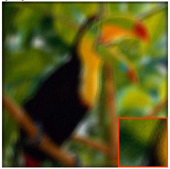
$\times 2$
(b) LR



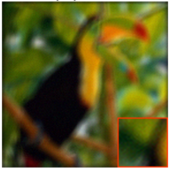
(21.33/0.5465)
(c) Bicubic



(21.25/0.5200)
(d) VDSR



(21.39/0.5323)
(e) TNRD



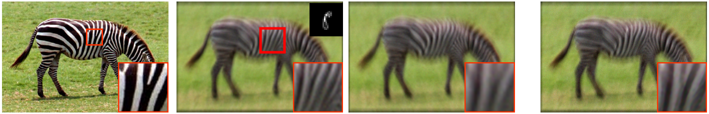
(21.23/0.5000)
(f) IRCNN

(19.61/0.4689)
(g) SRMD

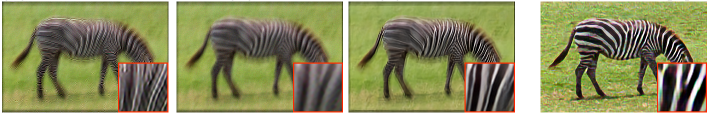
(26.06/0.6817)
(h) SRWDNet(ours)

Experimental Results IV

- Visual Results: $\times 3$ on Set14



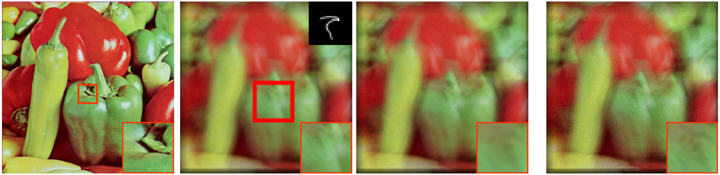
PSNR/SSIM (a) Ground-truth | $\times 3$ (b) LR | (16.04/0.2910) (c) Bicubic | (15.65/0.2547) (d) VDSR [8]



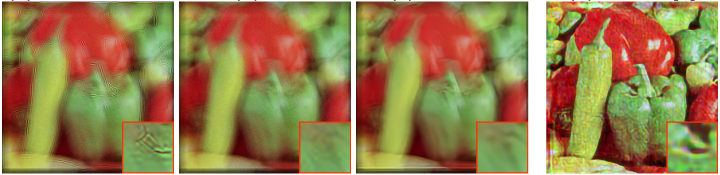
(15.70/0.3221) (e) TNRD [9] | (15.65/0.2516) (f) IRCNN [10] | (15.00/0.2509) (g) SRMD [11] | (20.25/0.4899) (h) SRWDNet(ours)

Experimental Results V

- Visual Results: $\times 4$ on Set14



PSNR/SSIM (a) Ground-truth | $\times 4$ (b) LR | (17.19/0.4903) (c) Bicubic | (17.16/0.4669) (d) VDSR [8]



(17.37/0.4817) (e) TNRD [9] | (17.16/0.4640) (f) IRCNN [10] | (17.15/0.4942) (g) SRMD [11] | (21.99/0.4707) (h) SRWDNet(ours)

Conclusion

- We propose an efficient deep SISR network to reconstruct sharp high-resolution images from blurred noisy low-resolution images.
- The proposed method uses the more realistic degradation model which is benefit existing non-blind deblurring methods for blur kernel estimation.
- We split the SISR problem into joint deblurring, denoising, and super-resolution tasks.
- We solve it by training the end-to-end network with the proximal gradient descent optimization in an iterative manner.

Questions?

Thank You

References I

Neal Parikh and Stephen Boyd.
Proximal algorithms.
Found. Trends Optim., 1(3):127–239, January 2014.

Stamatios Lefkimmiatis.
Universal denoising networks: A novel cnn architecture for image denoising.
2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3204–3213, 2018.

Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang.
Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.
2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1874–1883, 2016.

References II

Pablo Andres Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik.
Contour detection and hierarchical image segmentation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 33:898–916, 2011.

Radu Timofte, Vincent De Smet, and Luc Van Gool.
A+: Adjusted anchored neighborhood regression for fast super-resolution.
In *ACCV*, 2014.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja.
Single image super-resolution from transformed self-exemplars.
2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5197–5206, 2015.

References III

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.
2015 IEEE International Conference on Computer Vision (ICCV), pages 1026–1034, 2015.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee.
Accurate image super-resolution using very deep convolutional networks.
2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1646–1654, 2016.

References IV

Yunjin Chen and Thomas Pock.
Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 39:1256–1272, 2017.

Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang.
Learning deep cnn denoiser prior for image restoration.
2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2808–2817, 2017.

Kai Zhang, Wangmeng Zuo, and Lei Zhang.
Learning a single convolutional super-resolution network for multiple degradations.
2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3262–3271, 2018.