Deep Iterative Residual Convolutional Network for Single Image Super-Resolution

Rao Muhammad Umer, Gian Luca Foresti, Christian Micheloni

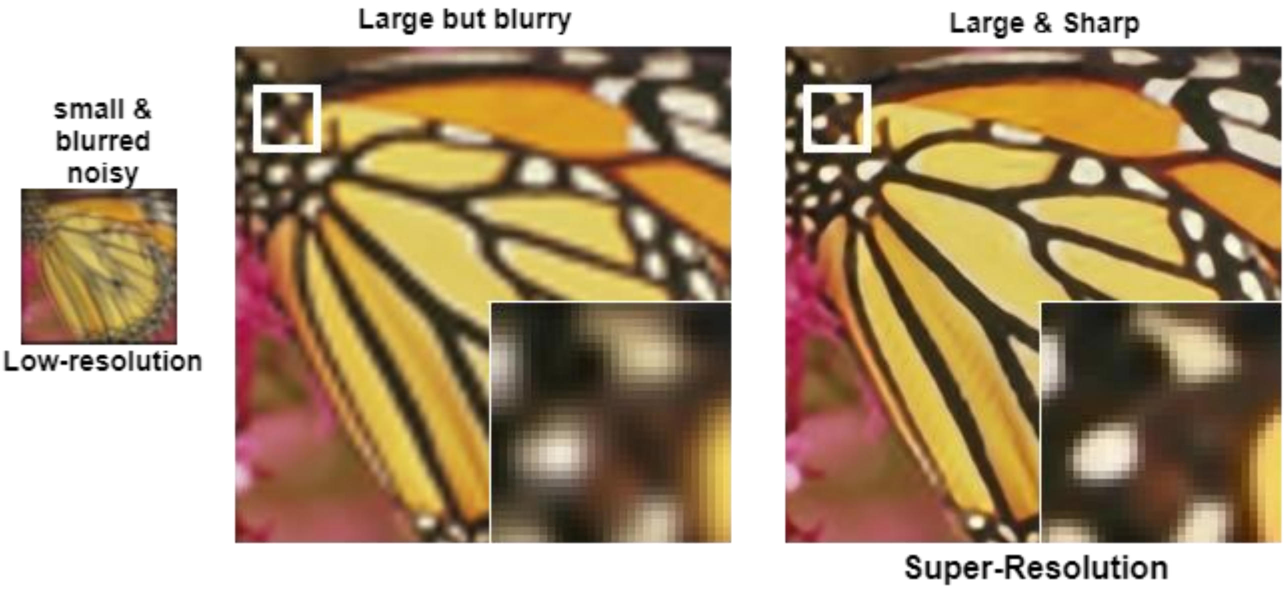
University of Udine, Udine, Italy.

GitHub

RaoUmer/ISRResCNet

SISR: Definition

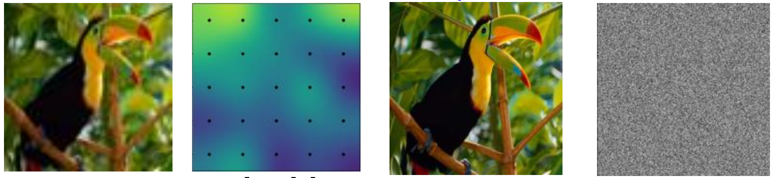
- **Single Image Super-Resolution (SISR):**
  - Restoration of the high-resolution (**HR**) image from its low-resolution (**LR**) counterpart.



Proposed Method

- **Problem Formulation:**
  - **Image Forward Observation model:**

$$\mathbf{y} = \mathbf{H}\tilde{\mathbf{x}} + \eta,$$



LR Bicubic HR Random Gaussian noise

  - **Inverse Problem:**
    - There could exist **multiple** possible **HR** images resulting in the **same downscaled LR** image.
    - **Regularization** is required in order to select the most plausible ones.

Proposed Method

- **Objective Function Minimization Strategy:**
  - Recovery of **x (unknown latent HR)** from **y (observed LR)** by the **variational approach**:

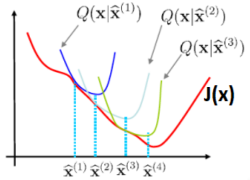
$$\mathbf{J}(\mathbf{x}) = \arg\min_{\mathbf{x}} \underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2}_{\text{Data fidelity}} + \lambda \underbrace{\mathcal{R}(\mathbf{x})}_{\text{Regularizer}},$$

(J(x): convex function but possibly **non-differentiable)**

measures the proximity of the solution to the observation

Image Priors

  - **Majorization-Minimization (MM)** framework:



$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathbf{J}(\mathbf{x}),$$

takes the form

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)}),$$

with

$$\mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)}) > \mathbf{J}(\mathbf{x}), \ \forall \mathbf{x} \neq \mathbf{x}^{(k)} \text{ and } \mathbf{Q}(\mathbf{x}^{(k)}; \mathbf{x}^{(k)}) = \mathbf{J}(\mathbf{x}^{(k)}).$$

  - **MM** variant by majorizing the quadratic data-term:

$$\mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda \mathbf{Q}_{\mathcal{R}}(\mathbf{x}; \mathbf{x}^{(k)}),$$

where

$$\mathbf{Q}_{\mathcal{R}}(\mathbf{x}; \mathbf{x}_0) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T[\alpha\mathbf{I} - \mathbf{H}^T\mathbf{H}](\mathbf{x} - \mathbf{x}_0),$$

$$\left.\begin{array}{l} \mathbf{G} = \alpha\mathbf{I} - \mathbf{H}^T\mathbf{H} \succ 0 : \text{ positive definite} \\ \lambda(\mathbf{G}) = \alpha - \lambda(\mathbf{H}^T\mathbf{H}) \end{array}\right\} \Rightarrow \alpha > \|\mathbf{H}^T\mathbf{H}\|$$

($\alpha \approx$1)

Proposed Method

- **Objective Function Minimization Strategy:**
  - **Overall majorizer:**

$$\mathbf{Q}(\mathbf{x};\mathbf{x}_0) = \frac{1}{2/\alpha}\|\mathbf{x}-\mathbf{z}\|_2^2 + \lambda\mathcal{R}(\mathbf{x}) + const.,$$

where

$$\mathbf{z} = \mathbf{x}_0 + \frac{1}{\alpha}\mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{x}_0),$$

  - **MM optimization** scheme to **iteratively minimize** the quadratic majorizer function **Q(.)** as:

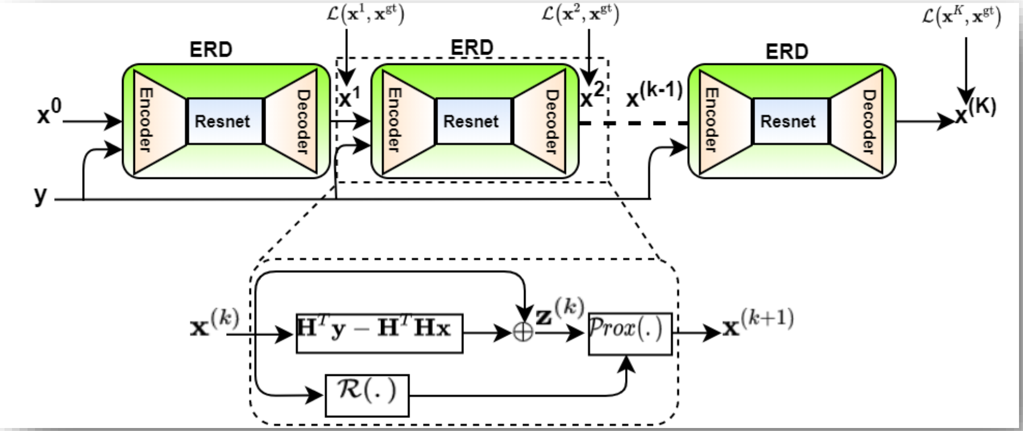
$$\begin{aligned}
\hat{\mathbf{x}}^{(k)} &= \arg\min_{\mathbf{x}} \mathbf{Q}(\mathbf{x};\mathbf{x}^k) \\
&= \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \lambda\mathbf{Q}_{\mathcal{R}}(\mathbf{x};\mathbf{x}^k) \\
&\boxed{= \arg\min_{\mathbf{x}} \frac{1}{2/\alpha}\|\mathbf{x}-\mathbf{z}^k\|_2^2 + \lambda\mathcal{R}(\mathbf{x})} \\
&= \mathrm{Prox}_{(\lambda/\alpha)\mathcal{R}(.)}(\mathbf{z}^k)
\end{aligned}$$

$$\mathbf{z}^k = \mathbf{z}^k + \mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{z}^k)$$

$$\mathbf{P}_{\mathcal{C}}(\mathbf{z}) = \arg\min_{\mathbf{x}\in\mathcal{C}} \frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{z}\|_2^2 + \frac{\lambda}{\alpha}\mathcal{R}(\mathbf{x}).$$

Proposed Method

- **Network Architecture:**
  - A **single optimizer** is used for all network stages with **shared structures** and **parameters** by **K steps**.

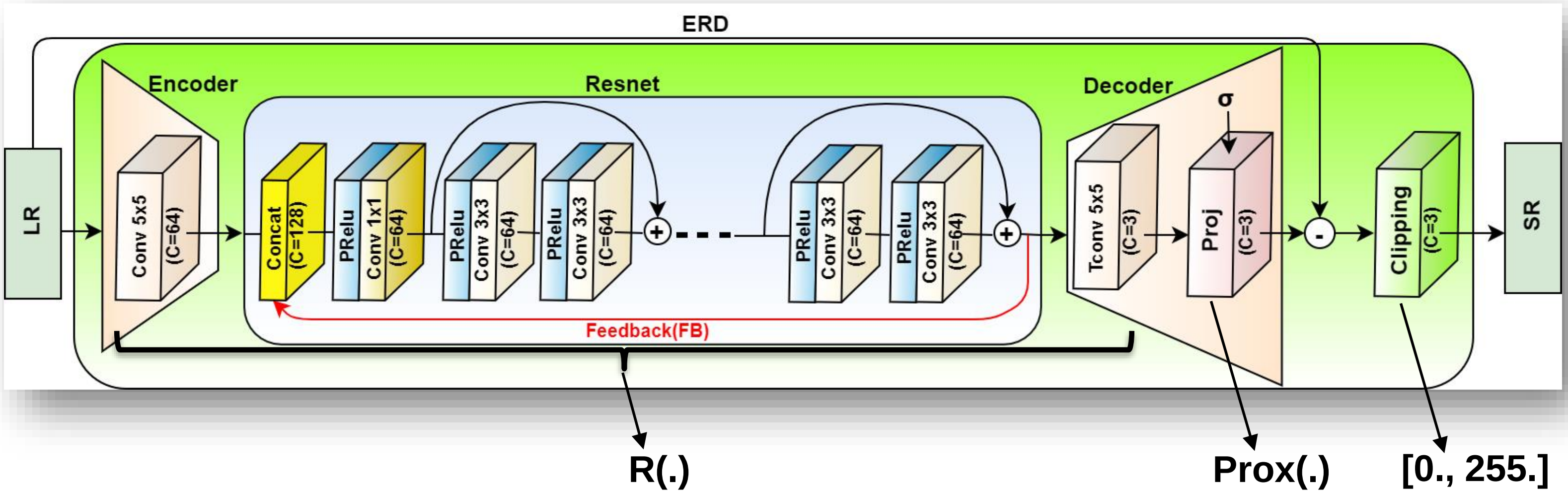


  - **Loss during training:**

$$\arg\min_{\Theta} \mathcal{L}(\Theta) = \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{x}_n^k - \mathbf{x}_n^{gt}\|_1$$

Proposed Method

- **ERD (Encoder-Resnet-Decoder) Block:**



Quantitative Results

- **Average PSNR/SSIM** values for scale factors **x2**, **x3**, and **x4** with **bicubic degradation model**. The best performance is shown in **red** and the second best performance is shown in **blue**.

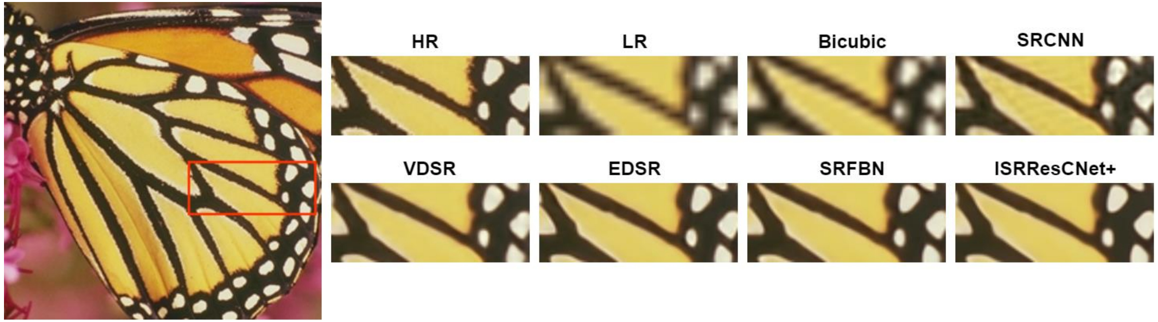
Dataset	Scale	Bicubic	SRCNN [5] (ECCV-2014)	VDSR [3] (CVPR-2016)	EDSR-baseline [4] (CVPR-2017)	RISR [7] (ICPR-2018)	SRFBN-S [9] (CVPR-2019)	ISRResCNet (Ours)	ISRResCNet+ (Ours)
Set5	×2	33.55 / 0.9304	36.16 / 0.9509	37.30 / 0.9573	37.59 / 0.9605	37.63 / 0.9590	37.39 / 0.9597	37.67 / 0.9596	37.79 / 0.9600
	×3	30.35 / 0.8686	32.28 / 0.9020	33.50 / 0.9197	34.18 / 0.9270	33.91 / 0.9234	33.99 / 0.9252	34.08 / 0.9251	34.20 / 0.9258
	×4	28.39 / 0.8109	29.99 / 0.8519	31.20 / 0.8818	31.89 / 0.8932	31.58 / 0.8870	31.76 / 0.8914	31.63 / 0.8890	31.77 / 0.8908
Set14	×2	30.05 / 0.8701	31.81 / 0.9033	32.84 / 0.9121	33.21 / 0.9177	33.16 / 0.9133	33.04 / 0.9157	32.89 / 0.9144	33.06 / 0.9155
	×3	27.40 / 0.7763	28.70 / 0.8151	29.54 / 0.8323	29.91 / 0.8421	29.91 / 0.8338	29.72 / 0.8376	29.63 / 0.8365	29.76 / 0.8381
	×4	25.86 / 0.7056	26.92 / 0.7427	27.75 / 0.7688	28.20 / 0.7820	28.19 / 0.7707	28.05 / 0.7785	27.99 / 0.7757	28.08 / 0.7776
B100	×2	29.51 / 0.8439	31.07 / 0.8838	31.83 / 0.8949	32.03 / 0.8996	32.01 / 0.8968	31.87 / 0.8972	31.98 / 0.8974	32.03 / 0.8980
	×3	27.19 / 0.7399	28.17 / 0.7799	28.80 / 0.7971	29.03 / 0.8056	28.92 / 0.7996	28.90 / 0.8015	28.91 / 0.8014	28.96 / 0.8024
	×4	25.96 / 0.6698	26.70 / 0.7029	27.27 / 0.7252	27.53 / 0.7365	27.37 / 0.7270	27.41 / 0.7321	27.40 / 0.7301	27.44 / 0.7313
Urban100	×2	26.84 / 0.8409	29.01 / 0.8885	30.67 / 0.9129	31.81 / 0.9271	31.06 / 0.9168	31.27 / 0.9208	31.29 / 0.9205	31.45 / 0.9220
	×3	24.44 / 0.7359	25.82 / 0.7874	27.09 / 0.8271	28.05 / 0.8524	27.41 / 0.8338	27.60 / 0.8418	27.57 / 0.8409	27.70 / 0.8432
	×4	23.13 / 0.6593	24.11 / 0.7051	25.14 / 0.7522	25.98 / 0.7850	25.41 / 0.7595	25.66 / 0.7725	25.56 / 0.7682	25.65 / 0.7705

Params: 1.5 M
Resblocks: 16

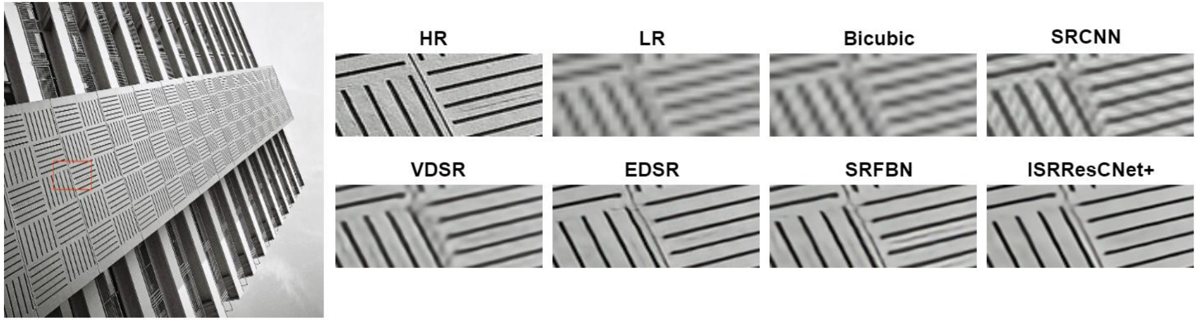
Params: 380 K
Resblocks: 5

Visual Results

- **SR results at the x4 scale on the test benchmark datasets:**



'butterfly' GT from Set5



'img_092' GT from Urban100

Conclusion

- Follows the **image observation (physical) model**.
- Solves the SISR problem in an **iterative manner** by minimizing the **discriminative loss** function with **residual learning**.
- Exploits the powerful image **regularization** and **large-scale optimization** techniques for image restoration.
- Achieves good PSNR/SSIM with **few trainable parameters**.
- Suitable for the **limited memory** storage and **CPU power** requirements for the **mobile/embedded** deployment.

Thank You!