Closed-loop Matters: Dual Regression Networks for Single Image Super-Resolution

Yong Guo, Jian Chen, Jingdong Wang, Qi Chen,
Jiezhang Cao, Zeshuai Deng, Yanwu Xu, Mingkui Tan

华南理工大学
South China University of Technology

Code: https://github.com/guoyongcs/DRN

Motivation

Limitations of existing SR methods

- The space of the possible functions that map LR to HR images is extremely large because infinitely many HR images can be downsampled to the same LR image
- It is hard to obtain a promising SR model when the paired data are unavailable

- It is hard for existing methods to find a good solution due to the large space of possible mapping functions
- SR models often incur a severe adaptation problem and yield poor performance on unpaired real-world data

Dual Regression Scheme

Our method

- We propose a novel dual regression scheme that can reduce the possible function space to enhance the performance of SR models

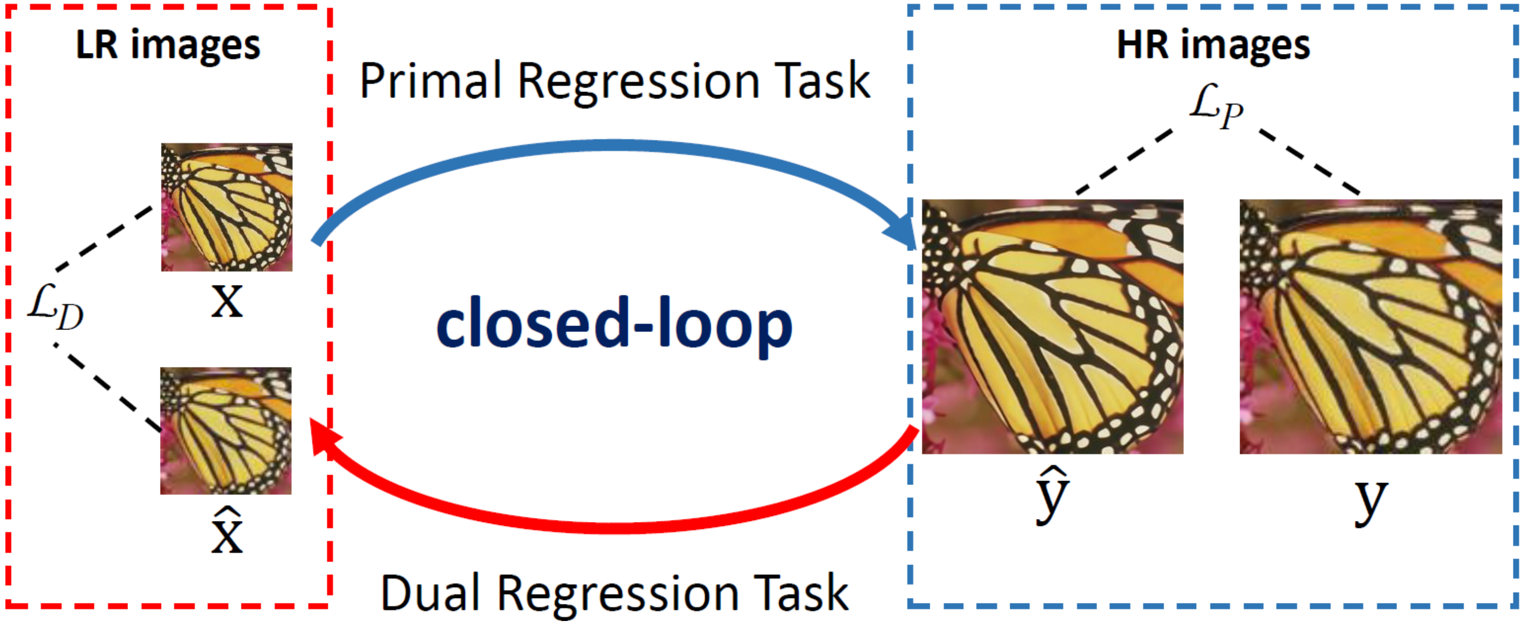


Figure 2. Dual regression training scheme, which contains a primal regression task for super-resolution and a dual regression task to project super-resolved images back to LR images

Dual Reconstruction Network (DRN)

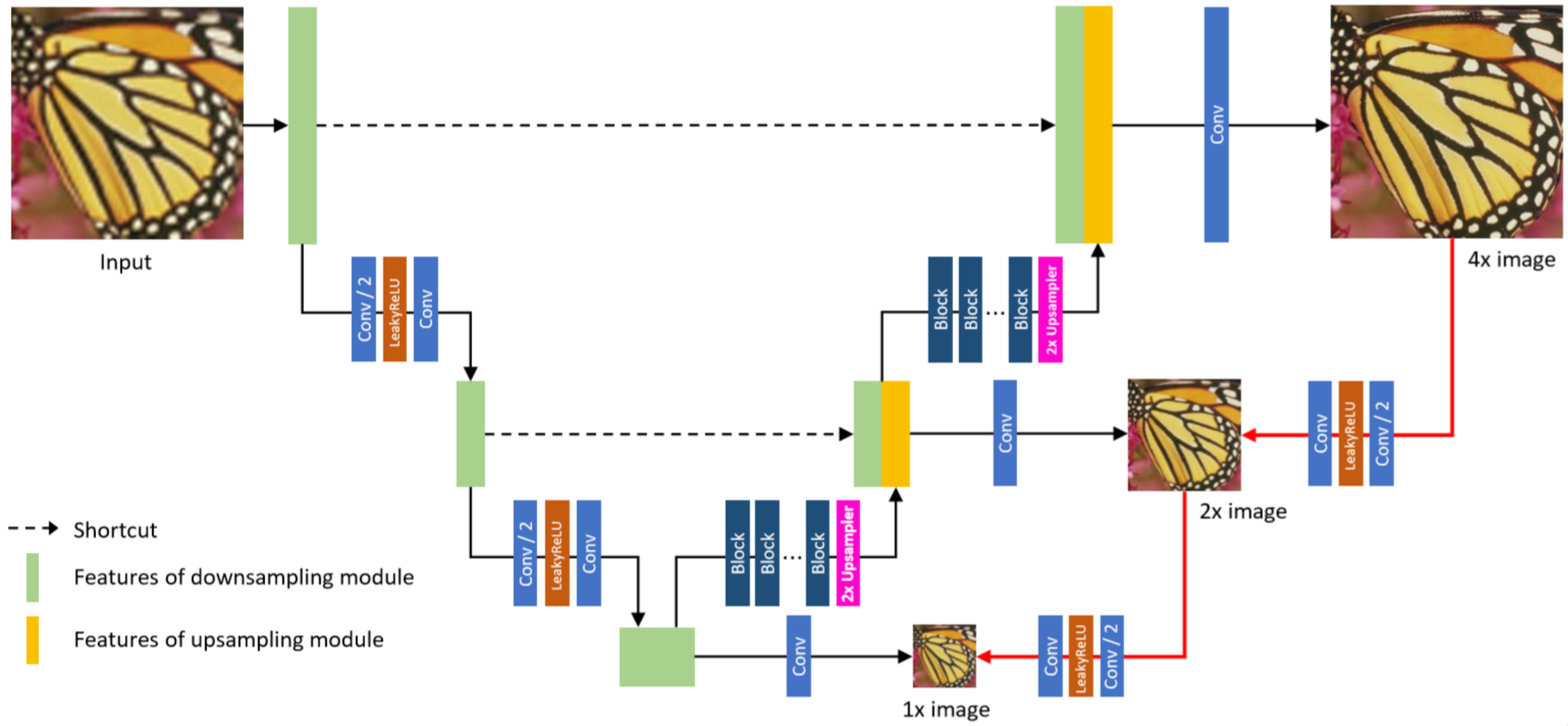


Figure 3. The architecture of DRN for 4× super-resolution

- DRN contains a primal network and a dual network (marked as red lines)
- The primal module follows the downsampling-upsampling design of U-Net
- The dual module has two convolution layers and a LeakyReLU activation layer

Training Methods for Paired Data

Given paired data, the model is trained by minimizing Eqn. (1) under the learning scheme of supervised SR methods:

$$\sum_{i=1}^{N} \underbrace{\mathcal{L}_P\Big(P(\mathbf{x}_i), \mathbf{y}_i\Big)}_{\text{primal regression loss}} + \lambda \underbrace{\mathcal{L}_D\Big(D(P(\mathbf{x}_i)), \mathbf{x}_i\Big)}_{\text{dual regression loss}} \qquad (1)$$

Notation

- $\boldsymbol{x}_i, \boldsymbol{y}_i$ donate the i-th pair of low- and high-resolution images
- $\mathcal{L}_P, \mathcal{L}_D$ are the loss function (L1-norm) for the primal and dual tasks
- λ controls the weight of the dual reconstruction loss

Adaptation Algorithm on Unpaired Data

Our method

- We propose an efficient adaptation algorithm to adapt SR models to the unpaired LR data

Given both unpaired data and paired data, the model is trained by minimizing Eqn. (2):

$$\sum_{i=1}^{M+N} \mathbf{1}_{\mathcal{S}_P}(\mathbf{x}_i)\mathcal{L}_P\Big(P(\mathbf{x}_i), \mathbf{y}_i\Big) + \lambda\mathcal{L}_D\Big(D(P(\mathbf{x}_i)), \mathbf{x}_i\Big) \quad (2)$$

Notation

- $1_{S_P}(\cdot)$ is an indicator function that equals to 1 when $\mathbf{x_i} \in S_P$ (S_P is paired dataset), and otherwise the function equals 0

Algorithm 1: Adaptation Algorithm on Unpaired Data.

Input: Unpaired real-world data: $\mathcal{S}_U$;
Paired synthetic data: $\mathcal{S}_P$;
Batch sizes for $\mathcal{S}_U$ and $\mathcal{S}_P$: m and n;
Indicator function: $\mathbf{1}_{\mathcal{S}_P}(\cdot)$.

Load the pretrained models P and D.
while *not convergent* **do**
  Sample unlabeled data $\{\mathbf{x}_i\}_{i=1}^{m}$ from $\mathcal{S}_U$;
  Sample labeled data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=m+1}^{m+n}$ from $\mathcal{S}_P$;
  // Update the primal model
  Update P by minimizing the objective:
  $$\sum_{i=1}^{m+n} \mathbf{1}_{\mathcal{S}_P}(\mathbf{x}_i)\mathcal{L}_P\Big(P(\mathbf{x}_i), \mathbf{y}_i\Big) + \lambda\mathcal{L}_D\Big(D(P(\mathbf{x}_i)), \mathbf{x}_i\Big)$$
  // Update the dual model
  Update D by minimizing the objective:
  $$\sum_{i=1}^{m+n} \lambda\mathcal{L}_D\Big(D(P(\mathbf{x}_i)), \mathbf{x}_i\Big)$$
end

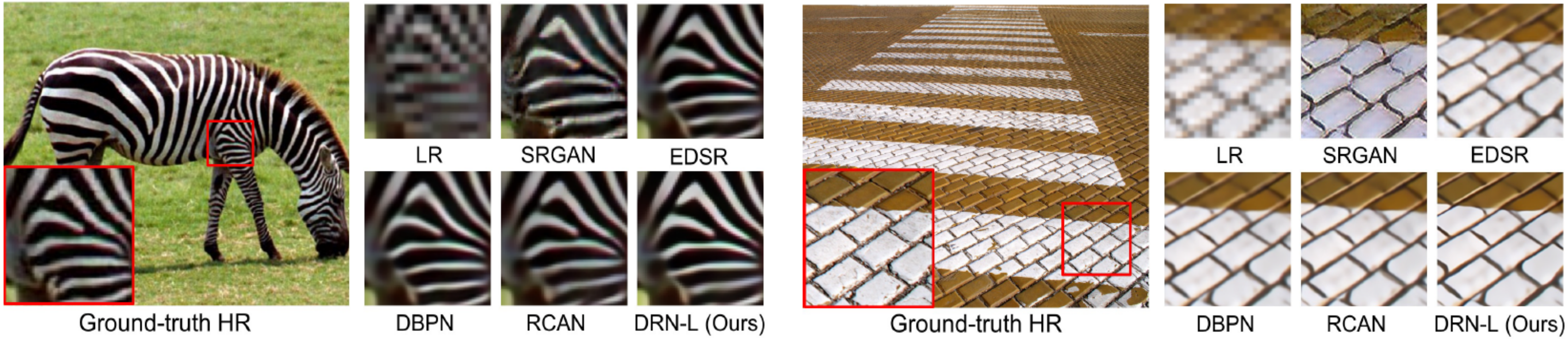
SR tasks with paired Bicubic data

Table 1. Performance comparison with state-of-the-art algorithms for 4× and 8× image super-resolution. The **bold** number indicates the best result and the **blue** number indicates the second best result. "-" denotes the results that are not reported.

Algorithms	Scale	#Params (M)	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSDS100 PSNR / SSIM	Urban100 PSNR / SSIM	Manga109 PSNR / SSIM
Bicubic	4	-	28.42 / 0.810	26.10 / 0.702	25.96 / 0.667	23.15 / 0.657	24.92 / 0.789
ESPCN [33]		-	29.21 / 0.851	26.40 / 0.744	25.50 / 0.696	24.02 / 0.726	23.55 / 0.795
SRResNet [24]		1.6	32.05 / 0.891	28.49 / 0.782	27.61 / 0.736	26.09 / 0.783	30.70 / 0.908
SRGAN [24]		1.6	29.46 / 0.838	26.60 / 0.718	25.74 / 0.666	24.50 / 0.736	27.79 / 0.856
LapSRN [23]		0.9	31.54 / 0.885	28.09 / 0.770	27.31 / 0.727	25.21 / 0.756	29.09 / 0.890
SRDenseNet [35]		2.0	32.02 / 0.893	28.50 / 0.778	27.53 / 0.733	26.05 / 0.781	29.49 / 0.899
EDSR [26]		43.1	32.48 / 0.898	28.81 / 0.787	27.72 / 0.742	26.64 / 0.803	31.03 / 0.915
DBPN [16]		10.4	32.42 / 0.897	28.75 / 0.786	27.67 / 0.739	26.38 / 0.794	30.90 / 0.913
RCAN [51]		15.6	32.63 / 0.900	28.85 / 0.788	27.74 / 0.743	26.74 / 0.806	31.19 / 0.917
SAN [8]		15.9	32.64 / 0.900	28.92 / 0.788	27.79 / 0.743	26.79 / 0.806	31.18 / 0.916
RRDB [37]		16.7	**32.73** / 0.901	**28.97** / 0.790	**27.83 / 0.745**	**27.02** / **0.815**	**31.64** / 0.919
DRN-S		4.8	32.68 / **0.901**	28.93 / **0.790**	27.78 / 0.744	26.84 / 0.807	31.52 / **0.919**
DRN-L		9.8	**32.74 / 0.902**	**28.98 / 0.792**	**27.83 / 0.745**	**27.03** / **0.813**	**31.73 / 0.922**
Bicubic	8	-	24.39 / 0.657	23.19 / 0.568	23.67 / 0.547	20.74 / 0.515	21.47 / 0.649
ESPCN [33]		-	25.02 / 0.697	23.45 / 0.598	23.92 / 0.574	21.20 / 0.554	22.04 / 0.683
SRResNet [24]		1.7	26.62 / 0.756	24.55 / 0.624	24.65 / 0.587	22.05 / 0.589	23.88 / 0.748
SRGAN [24]		1.7	23.04 / 0.626	21.57 / 0.495	21.78 / 0.442	19.64 / 0.468	20.42 / 0.625
LapSRN [23]		1.3	26.14 / 0.737	24.35 / 0.620	24.54 / 0.585	21.81 / 0.580	23.39 / 0.734
SRDenseNet [35]		2.3	25.99 / 0.704	24.23 / 0.581	24.45 / 0.530	21.67 / 0.562	23.09 / 0.712
EDSR [26]		45.5	27.03 / 0.774	25.05 / 0.641	24.80 / 0.595	22.55 / 0.618	24.54 / 0.775
DBPN [16]		23.2	27.25 / 0.786	25.14 / 0.649	24.90 / 0.602	22.72 / 0.631	25.14 / 0.798
RCAN [51]		15.7	27.31 / 0.787	25.23 / 0.651	24.96 / 0.605	**22.97 / 0.643**	25.23 / 0.802
SAN [8]		16.0	27.22 / 0.782	25.14 / 0.647	24.88 / 0.601	22.70 / 0.631	24.85 / 0.790
DRN-S		5.4	**27.41 / 0.790**	**25.25 / 0.652**	**24.98 / 0.605**	22.96 / 0.641	**25.30 / 0.805**
DRN-L		10.0	**27.43 / 0.792**	**25.28 / 0.653**	**25.00 / 0.606**	**22.99 / 0.644**	**25.33 / 0.806**

- DRN-S with about 5M parameters yields promising performance
- DRN-L with about 10M parameters yields the best performance

SR tasks with paired Bicubic data



(a) Visual comparison for 4× super-resolution

(b) Visual comparison for 8× super-resolution

Figure 4. Visual comparison of different methods for 4× and 8× image super-resolution

- Our model consistently produces images with sharper edges and shapes, while other baselines may give more blurry ones

SR tasks with unpaired data

Table 2. Adaptation performance of super-resolution models on images with different degradation methods for 8× SR.

Algorithms	Degradation	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSDS100 PSNR / SSIM	Urban100 PSNR / SSIM	Manga109 PSNR / SSIM
Nearest	Nearest	21.22 / 0.560	20.11 / 0.485	20.64 / 0.471	17.76 / 0.454	18.51 / 0.594
EDSR [26]		19.56 / 0.580	18.24 / 0.498	18.53 / 0.479	15.68 / 0.435	17.22 / 0.598
DBPN [16]		18.80 / 0.541	17.36 / 0.461	17.94 / 0.456	15.07 / 0.400	16.67 / 0.550
RCAN [51]		18.33 / 0.534	17.11 / 0.436	17.67 / 0.444	14.73 / 0.380	16.25 / 0.525
CinCGAN [43]		21.76 / 0.648	20.64 / 0.552	20.89 / 0.528	18.21 / 0.505	18.86 / **0.638**
DRN-Adapt		**23.00 / 0.715**	**21.52 / 0.561**	**21.98 / 0.539**	**19.07 / 0.518**	**19.83** / 0.613
EDSR [26]	BD	23.54 / 0.702	22.13 / 0.594	22.71 / 0.567	19.70 / 0.551	20.64 / 0.700
DBPN [16]		23.05 / 0.693	21.65 / 0.586	22.50 / 0.565	19.28 / 0.538	20.16 / 0.689
RCAN [51]		22.23 / 0.678	21.01 / 0.567	21.85 / 0.552	18.36 / 0.509	19.34 / 0.659
CinCGAN [43]		23.39 / 0.682	22.14 / 0.581	22.73 / 0.554	20.36 / 0.538	20.29 / 0.670
DRN-Adapt		**24.62 / 0.719**	**23.07 / 0.612**	**23.59 / 0.583**	**20.57 / 0.591**	**21.52 / 0.714**

- DRN-Adapt outperforms the baseline methods on unpaired synthetic data

Figure 5. Visual comparison of model adaptation to real-world video frames (from YouTube) for 8× SR

- DRN-Adapt produces visually promising images with sharper and clearer textures

Conclusion

- We propose a theoretically guaranteed dual regression scheme that can reduce the possible function space to enhance the performance of SR models
- We propose an efficient adaptation algorithm to adapt SR model to unpaired real-world data, such as raw video frames from YouTube
- Extensive experiments on both paired and unpaired data demonstrate the superiority of DRN over the considered baseline methods

Code: https://github.com/guoyongcs/DRN