Alma Mater Studiorum
University of Bologna

Artificial Intelligence
Machine Learning for Computer Vision
Alessandro Dicosola [Matr. 935563]

Super Resolution

Survey Any-Scale Deep Network (ASDN [1])

Contents

1 Introduction

In computer vision the **super resolution** (hereinafter SR) task is an ill-posed problem for reconstructing a low resolution (hereinafter LR) image to an higher one (hereinafter SR - super resolution image); in order to do so an high resolution image is used (hereinafter HR).

Many method proposed trying to define the images in the features space and apply linear and non-linear operators in order to learn and reconstruct the high resolution image: using random forest, linear and non-linear regressor, manifold embedding[8].

Due to advancement of deep learning, SR task started to be achieved using deep neural networks: from the simplest one, SRCNN[9] with only three convolution for extracting and processing features used then for reconstructing the SR image to more complex ones that use short,long and dense skip connections [10][2] and recursive blocks [11][12], attention mechanism [3][4], meta-learning [7], multi-levels [5][6].

Studies done in deep learning has lead research to understand that deeper and wider networks performs better than shallow and narrow ones as well as using residual learning; skip connections (short,long and dense) are used for overcoming the exploding/vanishing gradient problem and at the same time they allow to let networks converge faster; dense connection allow also to reuse features from previous stages and attention mechanism allow networks to focus on most important features (channel-wise and spatial-wise).

SR task achieved with deep learning trying to learn a mapping between LR to HR images given a specific scale. The studies done up to now always used fixed integer scales (2x,3x,4x,...) but none had explored decimal scales but Meta-SR[7] (using meta-learning).

Therefore **Any-Scale Deep Network** (ASDN) tried to explore decimal scale in SR task learning the mapping in an end-to-end way.

2 Background

Here will be explored the most important networks referenced by ASDN [1].

2.1 Deep Bi-Dense Network (DBDN[2])

2.2 Residual Channel Attention Network (RCAN[3])

2.3 Channel-wise and Spatial Feature Modulation Network (CSFM[4])

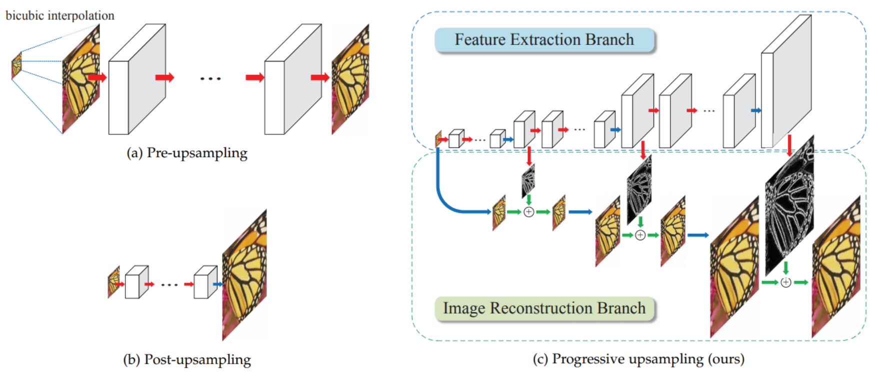

Fig. 1. **Comparisons of upsampling strategies in CNN-based SR algorithms.** Red arrows indicate convolutional layers. Blue arrows indicate transposed convolutions (upsampling), and green arrows denote element-wise addition operators. (a) Pre-upsampling based approaches (e.g., SRCNN [9], VDSR [11], DRCN [12], DRRN [13]) typically use the bicubic interpolation to upscale LR input images to the target spatial resolution before applying deep networks for prediction and reconstruction. (b) Post-upsampling based methods directly extract features from LR input images and use sub-pixel convolution [14] or transposed convolution [15] for upsampling. (c) Progressive upsampling approach using the proposed Laplacian pyramid network reconstructs HR images in a coarse-to-fine manner.

Figure 1: LapSRN architecture.

2.4 Deep Laplacian Pyramid Network (LapSRN[5]) and Multi-scale Deep Laplacian Pyramid Network (MS-LapSRN [6])

2.4.1 Features

LapSRN in order to be able to reconstruct the SR image uses:

- **Charbonnier loss** because the L2 loss is not able to capture the underlying mapping of LR images to many HR images.
- **progressive reconstruction** of the SR image using the Laplacian Pyramid Framework [13].
- **residual learning**: learn the summation between the laplacian extracted by *features extraction branch* and the upscaled LR image in the *image reconstruction branch*

MS-LapSRN improve LapSRN introducing:

- **parameter sharing** across pyramid level and within pyramid levels in order to reduce the amount of parameters which increase with the scale (the greater the scale the deeper the network since there are more levels).
- **local skip connections** for avoiding the vanishing/exploding gradient prolem with the increase in the depth of the network.
- **multi scale training**: the previous network was trained for each scale different networks.

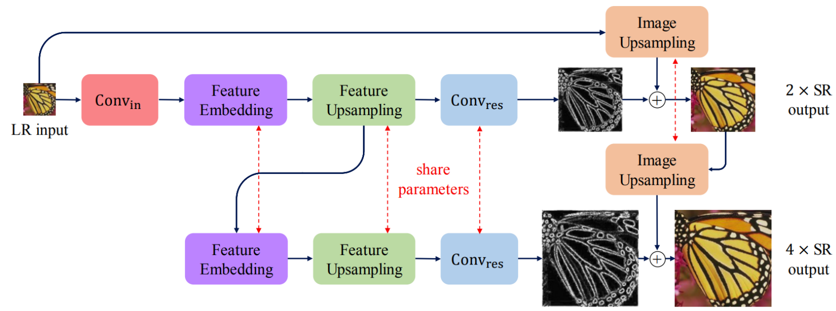

Fig. 3. **Detailed network architecture of the proposed LapSRN.** At each pyramid level, our model consists of a feature embedding sub-network for extracting non-linear features, transposed convolutional layers for upsampling feature maps and images, and a convolutional layer for predicting the sub-band residuals. As the network structure at each level is highly similar, we share the weights of those components across pyramid levels to reduce the number of network parameters.

Figure 2: MS-LapSRN architecture.

2.4.2 Architecture

In *LapSRN* [Figure 1] the **feature extraction branch** extract laplacian representations which are used for training (in an end-to-end fashion) the **image reconstruction branch** in order to reconstruct the SR image at the last level (due to the laplacian pyramid framework training on an higher scales lead to have also a network capable to resolve SR task for lower scale).

The residual learning allow the network to focus on high-frequency information (edges) instead of low-frequency ones.

In the *MS-LapSRN* [Figure 2] the feature extractor ($Conv_{in}$, Feature Embedding, Feature Upsampling, $Conv_{res}$) has always the same function: extract meaningful information from an input image in order to create a residual whose spatial dimension in 2x then the input one; therefore is logic to use same weights for doing so.

The *Feature embedding* has **R** recursive block [11] [12] which contains distinct **D** convolutional layers.

The structure of the recursive block [Figure 3b] use a skip connection directly connected to the input and a pre-activation inside the residual path [14].

2.4.3 Loss

Both *LapSRN* and *MS-LapSRN* use the **Charbonnier loss**: the former use it once the latter at each level; the equation is the following:

$$L_S = \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L}\rho \times \left(\left(y_l^{(i)} - x_l^{(i)}\right) - r_l^{(i)}\right)$$

where S is the target upsampling factor, $\rho = \sqrt{x^2+\epsilon^2}$ (Charbonnier penalty function which is a differentiable L1 norm), x_l is the upscaled LR image at level

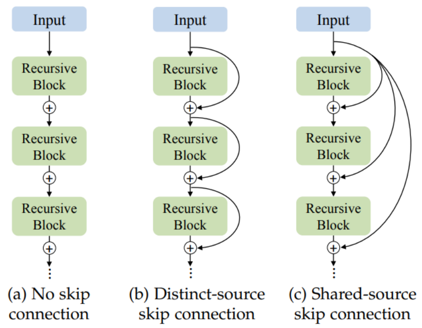

Fig. 4. **Local residual learning.** We explore three different ways of local skip connection in the feature embedding sub-network of the LapSRN for training deeper models.

(a) Differences between local skip connections.

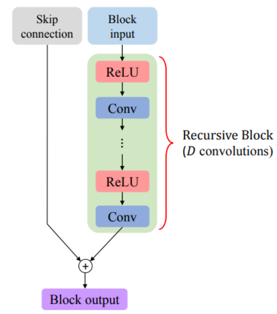

Fig. 5. **Structure of our recursive block.** There are D convolutional layers in a recursive block. The weights of convolutional layers are distinct within the block but shared among all recursive blocks. We use the pre-activation structure [41] without the batch normalization layer.

(b) Recursive block used in MS-LapSRN.

l, y_l is the downscaled HR target image (y) at level l and r_l is the residual at level l.

The final loss is the sum of the loss for each level trained using images with different scales (**multi scales training**):

$$L = \sum_{s \in [2,4,8]} L_s$$

2.4.4 Results

Here will be presented only the results of **MS-LapSRN**[6] since it's an evolution of LapSRN.

Ablation studies on pyramid structure, global residual learning and loss From the performance studies [Figure 4] it's possible to observe the contribution of each component used for the final result: the pyramid structures allow a faster convergence, the Charbonnier loss a better performance, the global residual learning both.

Studies on local skip connections From the local skip connection studies [Figure 5] we can se that the best is *shared source* (**SS**) which use as identity the same input.

Studies on parameter sharing From the parameters studies [Figure 6] it's possible to see the effectiveness of using recursive block which allow to have a deeper network with an increase of performance without too much parameters and also a faster inference time.

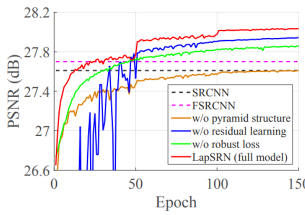

Fig. 6. **Convergence analysis**. We analyze the contributions of the pyramid structures, loss functions, and global residual learning by replacing each component with the one used in existing methods. Our full model converges faster and achieves better performance.

(a) Study on performance with different settings.

TABLE 2
Ablation study of LapSRN. Our full model performs favorably against several variants of the LapSRN on both SET5 and SET14 for $4\times$ SR.

GRL	Pyramid	Loss	SET5	SET14
✓		Charbonnier	30.58	27.61
	✓	Charbonnier	31.10	27.94
✓	✓	$\mathcal{L}_2$	30.93	27.86
✓	✓	Charbonnier	**31.28**	**28.04**

(b) Ablation study on LapSRN.

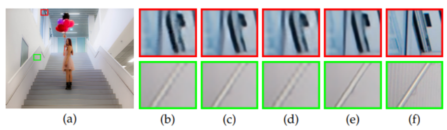

Fig. 7. **Contribution of different components in LapSRN**. (a) Ground truth HR image (b) without pyramid structure (c) without global residual learning (d) without robust loss (e) full model (f) HR patch.

(c) Contribution of different components in LapSRN.

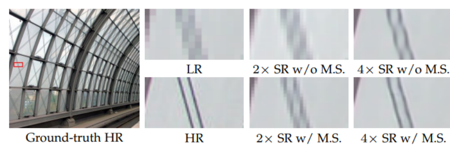

Fig. 8. **Contribution of multi-scale supervision (M.S.)**. The multi-scale supervision guides the network training to progressively reconstruct the HR images and help reduce the spatial aliasing artifacts.

(d) Contribution of multi-scale supervision.

Figure 4: Performance studies done on MS-LapSRN

Quantitative and qualitative results Fro the quantitative and qualitative results [Figure 7] we can see that overall LapSRN performs better than state-of-the-art network, indeed it is able to reconstruct finer details thanks to the multi-scale training as we can see in the figure.

Quantitative evaluation of local residual learning. We compare three different local residual learning methods on the URBAN100 dataset for $4\times$ SR. Overall, the shared local skip connection method (LapSRN_{SS}) achieves superior performance for deeper models.

Model	Depth	LapSRN_{NS}	LapSRN_{DS}	LapSRN_{SS}
D5R2	24	25.16	25.22	**25.23**
D5R5	54	25.18	25.33	**25.34**
D5R8	84	25.26	25.33	**25.38**

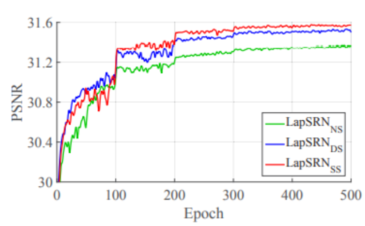

Fig. 9. **Comparisons of local residual learning**. We train our LapSRN-D5R5 model with three different local residual learning methods as described in Section 3.2.3 and evaluate on the SET5 for $4\times$ SR.

Figure 5: Local skip connection studies done on MS-LapSRN

Quantitative evaluation of the number of recursive blocks R and the number of convolutional layers D in our feature embedding sub-network. We build LapSRN with different network depth by varying the values of D and R and evaluate on the BSDS100 and URBAN100 datasets for 4× SR.

Model	#Parameters	Depth	BSDS100	URBAN100
D2R5	112k	24	27.33	25.24
D2R12	112k	52	27.35	**25.31**
D2R20	112k	84	**27.37**	**25.31**
D4R3	185k	28	27.33	25.25
D4R6	185k	52	**27.37**	25.34
D4R10	185k	84	**27.37**	**25.35**
D5R2	222k	24	27.32	25.23
D5R5	222k	54	27.38	25.34
D5R8	222k	84	**27.39**	**25.38**
D10R1	407k	24	27.33	25.23
D10R2	407k	44	27.36	25.27
D10R4	407k	84	**27.38**	**25.36**

Parameter sharing in LapSRN. We reduce the number of network parameters by sharing the weights between pyramid levels and applying recursive layers in the feature embedding sub-network.

Model	#Parameters	BSDS100	URBAN100
LapSRN [16]	812k	**27.32**	**25.21**
LapSRN_{NS}-D10R1	407k	**27.32**	25.20
LapSRN_{NS}-D5R2	222k	27.30	25.16
LapSRN_{NS}-D2R5	112k	27.26	25.10

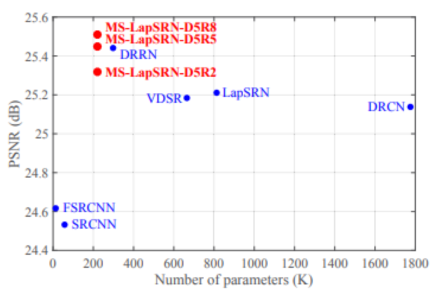

Fig. 15. **Number of network parameters versus performance**. The results are evaluated on the URBAN100 dataset for 4× SR. The proposed MS-LapSRN strides a balance between reconstruction accuracy and execution time.

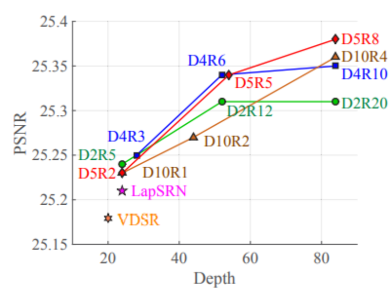

Fig. 10. **PSNR versus network depth**. We test the proposed model with different D and R on the URBAN100 dataset for 4× SR.

Figure 6: Studies on parameters (weights) and hyperparameters D and R on MS-LapSRN

Quantitative evaluation of state-of-the-art SR algorithms. We report the average PSNR/SSIM/IFC for $2\times$, $3\times$, $4\times$ and $8\times$ SR. Red and blue indicate the best and the second best performance, respectively. Both LapSRN [16] and the proposed MS-LapSRN do not use any $3\times$ SR images for training. To generate the results of $3\times$ SR, we first perform $4\times$ SR on input LR images and then downsample the output to the target resolution.

Algorithm	Scale	SET5 PSNR / SSIM / IFC	SET14 PSNR / SSIM / IFC	BSDS100 PSNR / SSIM / IFC	URBAN100 PSNR / SSIM / IFC	MANGA109 PSNR / SSIM / IFC
Bicubic		33.69 / 0.931 / 6.166	30.25 / 0.870 / 6.126	29.57 / 0.844 / 5.695	26.89 / 0.841 / 6.319	30.86 / 0.936 / 6.214
A+ [3]		36.60 / 0.955 / 8.715	32.32 / 0.906 / 8.200	31.24 / 0.887 / 7.464	29.25 / 0.895 / 8.440	35.37 / 0.968 / 8.906
RFL [5]		36.59 / 0.954 / 8.741	32.29 / 0.905 / 8.224	31.18 / 0.885 / 7.473	29.14 / 0.891 / 8.439	35.12 / 0.966 / 8.921
SelfExSR [22]		36.60 / 0.955 / 8.404	32.24 / 0.904 / 8.018	31.20 / 0.887 / 7.239	29.55 / 0.898 / 8.414	35.82 / 0.969 / 8.721
SRCNN [9]		36.72 / 0.955 / 8.166	32.51 / 0.908 / 7.867	31.38 / 0.889 / 7.242	29.53 / 0.896 / 8.092	35.76 / 0.968 / 8.471
FSRCNN [15]	$2\times$	37.05 / 0.956 / 8.199	32.66 / 0.909 / 7.841	31.53 / 0.892 / 7.180	29.88 / 0.902 / 8.131	36.67 / 0.971 / 8.587
SCN [10]		36.58 / 0.954 / 7.358	32.35 / 0.905 / 7.085	31.26 / 0.885 / 6.500	29.52 / 0.897 / 7.324	35.51 / 0.967 / 7.601
VDSR [11]		37.53 / 0.959 / 8.190	33.05 / 0.913 / 7.878	31.90 / 0.896 / 7.169	30.77 / 0.914 / 8.270	37.22 / 0.975 / 9.120
DRCN [12]		37.63 / 0.959 / 8.326	33.06 / 0.912 / 8.025	31.85 / 0.895 / 7.220	30.76 / 0.914 / 8.527	37.63 / 0.974 / 9.541
LapSRN [16]		37.52 / 0.959 / 9.010	33.08 / 0.913 / 8.501	31.80 / 0.895 / 7.715	30.41 / 0.910 / 8.907	37.27 / 0.974 / 9.481
DRRN [13]		37.74 / 0.959 / 8.671	33.23 / 0.914 / 8.320	32.05 / 0.897 / 7.613	31.23 / 0.919 / 8.917	37.92 / 0.976 / 9.268
MS-LapSRN-D5R2 (ours)		37.62 / 0.960 / 9.038	33.13 / 0.913 / 8.539	31.93 / 0.897 / 7.776	30.82 / 0.915 / 9.081	37.38 / 0.975 / 9.434
MS-LapSRN-D5R5 (ours)		37.72 / 0.960 / 9.265	33.24 / 0.914 / 8.726	32.00 / 0.898 / 7.906	31.01 / 0.917 / 9.334	37.71 / 0.975 / 9.710
MS-LapSRN-D5R8 (ours)		37.78 / 0.960 / 9.305	33.28 / 0.915 / 8.748	32.05 / 0.898 / 7.927	31.15 / 0.919 / 9.406	37.78 / 0.976 / 9.765
Bicubic		30.41 / 0.869 / 3.596	27.55 / 0.775 / 3.491	27.22 / 0.741 / 3.168	24.47 / 0.737 / 3.661	26.99 / 0.859 / 3.521
A+ [3]		32.62 / 0.909 / 4.979	29.15 / 0.820 / 4.545	28.31 / 0.785 / 4.028	26.05 / 0.799 / 4.883	29.93 / 0.912 / 4.880
RFL [5]		32.47 / 0.906 / 4.956	29.07 / 0.818 / 4.533	28.23 / 0.782 / 4.023	25.88 / 0.792 / 4.781	29.61 / 0.905 / 4.758
SelfExSR [22]		32.66 / 0.910 / 4.911	29.18 / 0.821 / 4.505	28.30 / 0.786 / 3.923	26.45 / 0.810 / 4.988	27.57 / 0.821 / 2.193
SRCNN [9]		32.78 / 0.909 / 4.682	29.32 / 0.823 / 4.372	28.42 / 0.788 / 3.879	26.25 / 0.801 / 4.630	30.59 / 0.914 / 4.698
FSRCNN [15]	$3\times$	33.18 / 0.914 / 4.970	29.37 / 0.824 / 4.569	28.53 / 0.791 / 4.061	26.43 / 0.808 / 4.878	31.10 / 0.921 / 4.912
SCN [10]		32.62 / 0.908 / 4.321	29.16 / 0.818 / 4.006	28.33 / 0.783 / 3.553	26.21 / 0.801 / 4.253	30.22 / 0.914 / 4.302
VDSR [11]		33.67 / 0.921 / 5.088	29.78 / 0.832 / 4.606	28.83 / 0.799 / 4.043	27.14 / 0.829 / 5.045	32.01 / 0.934 / 5.389
DRCN [12]		33.83 / 0.922 / 5.202	29.77 / 0.832 / 4.686	28.80 / 0.797 / 4.070	27.15 / 0.828 / 5.187	32.31 / 0.936 / 5.564
LapSRN [16]		33.82 / 0.922 / 5.194	29.87 / 0.832 / 4.662	28.82 / 0.798 / 4.057	27.07 / 0.828 / 5.168	32.21 / 0.935 / 5.406
DRRN [13]		34.03 / 0.924 / 5.397	29.96 / 0.835 / 4.878	28.95 / 0.800 / 4.269	27.53 / 0.764 / 5.456	32.74 / 0.939 / 5.659
MS-LapSRN-D5R2 (ours)		33.88 / 0.923 / 5.165	29.89 / 0.834 / 4.637	28.87 / 0.800 / 4.040	27.23 / 0.831 / 5.142	32.28 / 0.936 / 5.384
MS-LapSRN-D5R5 (ours)		34.01 / 0.924 / 5.307	29.96 / 0.836 / 4.758	28.92 / 0.801 / 4.127	27.39 / 0.835 / 5.333	32.60 / 0.938 / 5.559
MS-LapSRN-D5R8 (ours)		34.06 / 0.924 / 5.390	29.97 / 0.836 / 4.806	28.93 / 0.802 / 4.154	27.47 / 0.837 / 5.409	32.68 / 0.939 / 5.621
Bicubic		28.43 / 0.811 / 2.337	26.01 / 0.704 / 2.246	25.97 / 0.670 / 1.993	23.15 / 0.660 / 2.386	24.93 / 0.790 / 2.289
A+ [3]		30.32 / 0.860 / 3.260	27.34 / 0.751 / 2.961	26.83 / 0.711 / 2.565	24.34 / 0.721 / 3.218	27.03 / 0.851 / 3.177
RFL [5]		30.17 / 0.855 / 3.205	27.24 / 0.747 / 2.924	26.76 / 0.708 / 2.538	24.20 / 0.712 / 3.101	26.80 / 0.841 / 3.055
SelfExSR [22]		30.34 / 0.862 / 3.249	27.41 / 0.753 / 2.952	26.84 / 0.713 / 2.512	24.83 / 0.740 / 3.381	27.83 / 0.866 / 3.358
SRCNN [9]		30.50 / 0.863 / 2.997	27.52 / 0.753 / 2.766	26.91 / 0.712 / 2.412	24.53 / 0.725 / 2.992	27.66 / 0.859 / 3.045
FSRCNN [15]	$4\times$	30.72 / 0.866 / 2.994	27.61 / 0.755 / 2.722	26.98 / 0.715 / 2.370	24.62 / 0.728 / 2.916	27.90 / 0.861 / 2.950
SCN [10]		30.41 / 0.863 / 2.911	27.39 / 0.751 / 2.651	26.88 / 0.711 / 2.309	24.52 / 0.726 / 2.860	27.39 / 0.857 / 2.889
VDSR [11]		31.35 / 0.883 / 3.496	28.02 / 0.768 / 3.071	27.29 / 0.726 / 2.627	25.18 / 0.754 / 3.405	28.83 / 0.887 / 3.664
DRCN [12]		31.54 / 0.884 / 3.502	28.03 / 0.768 / 3.066	27.24 / 0.725 / 2.587	25.14 / 0.752 / 3.412	28.98 / 0.887 / 3.674
LapSRN [16]		31.54 / 0.885 / 3.559	28.19 / 0.772 / 3.147	27.32 / 0.727 / 2.677	25.21 / 0.756 / 3.530	29.09 / 0.890 / 3.729
DRRN [13]		31.68 / 0.888 / 3.703	28.21 / 0.772 / 3.252	27.38 / 0.728 / 2.760	25.44 / 0.764 / 3.700	29.46 / 0.896 / 3.878
MS-LapSRN-D5R2 (ours)		31.62 / 0.887 / 3.585	28.16 / 0.772 / 3.151	27.36 / 0.729 / 2.684	25.32 / 0.760 / 3.537	29.18 / 0.892 / 3.750
MS-LapSRN-D5R5 (ours)		31.74 / 0.888 / 3.705	28.25 / 0.773 / 3.238	27.42 / 0.731 / 2.737	25.45 / 0.765 / 3.674	29.48 / 0.896 / 3.888
MS-LapSRN-D5R8 (ours)		31.74 / 0.889 / 3.749	28.26 / 0.774 / 3.261	27.43 / 0.731 / 2.755	25.51 / 0.768 / 3.727	29.54 / 0.897 / 3.928
Bicubic		24.40 / 0.658 / 0.836	23.10 / 0.566 / 0.784	23.67 / 0.548 / 0.646	20.74 / 0.516 / 0.858	21.47 / 0.650 / 0.810
A+ [3]		25.53 / 0.693 / 1.077	23.89 / 0.595 / 0.983	24.21 / 0.569 / 0.797	21.37 / 0.546 / 1.092	22.39 / 0.681 / 1.056
RFL [5]		25.38 / 0.679 / 0.991	23.79 / 0.587 / 0.916	24.13 / 0.563 / 0.749	21.27 / 0.536 / 0.992	22.28 / 0.669 / 0.968
SelfExSR [22]		25.49 / 0.703 / 1.121	23.92 / 0.601 / 1.005	24.19 / 0.568 / 0.773	21.81 / 0.577 / 1.283	22.99 / 0.719 / 1.244
SRCNN [9]		25.33 / 0.690 / 0.938	23.76 / 0.591 / 0.865	24.13 / 0.566 / 0.705	21.29 / 0.544 / 0.947	22.46 / 0.695 / 1.013
FSRCNN [15]	$8\times$	25.60 / 0.697 / 1.016	24.00 / 0.599 / 0.942	24.31 / 0.572 / 0.767	21.45 / 0.550 / 0.995	22.72 / 0.692 / 1.009
SCN [10]		25.59 / 0.706 / 1.063	24.02 / 0.603 / 0.967	24.30 / 0.573 / 0.777	21.52 / 0.560 / 1.074	22.68 / 0.701 / 1.073
VDSR [11]		25.93 / 0.724 / 1.199	24.26 / 0.614 / 1.067	24.49 / 0.583 / 0.859	21.70 / 0.571 / 1.199	23.16 / 0.725 / 1.263
DRCN [12]		25.93 / 0.723 / 1.192	24.25 / 0.614 / 1.057	24.49 / 0.582 / 0.854	21.71 / 0.571 / 1.197	23.20 / 0.724 / 1.257
LapSRN [16]		26.15 / 0.738 / 1.302	24.35 / 0.620 / 1.133	24.54 / 0.586 / 0.893	21.81 / 0.581 / 1.288	23.39 / 0.735 / 1.352
DRRN [13]		26.18 / 0.738 / 1.307	24.42 / 0.622 / 1.127	24.59 / 0.587 / 0.891	21.88 / 0.583 / 1.299	23.60 / 0.742 / 1.406
MS-LapSRN-D5R2 (ours)		26.20 / 0.747 / 1.366	24.45 / 0.626 / 1.170	24.61 / 0.590 / 0.920	21.95 / 0.592 / 1.364	23.70 / 0.751 / 1.470
MS-LapSRN-D5R5 (ours)		26.34 / 0.752 / 1.414	24.57 / 0.629 / 1.200	24.65 / 0.591 / 0.938	22.06 / 0.597 / 1.426	23.85 / 0.756 / 1.538
MS-LapSRN-D5R8 (ours)		26.34 / 0.753 / 1.435	24.57 / 0.629 / 1.209	24.65 / 0.592 / 0.943	22.06 / 0.598 / 1.446	23.90 / 0.759 / 1.564

(a) SOtA results up to now

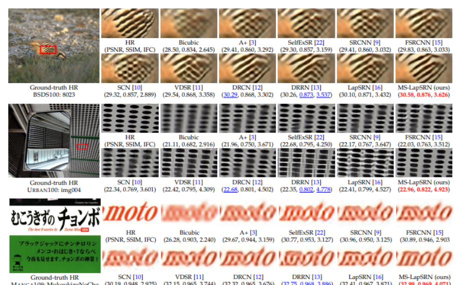

Fig. 11. **Visual comparison for $4\times$ SR on the BSDS100. URBAN100 and MANGA109 datasets.**

(b) Qualitative results of MS-LapSRN.

(c) $8\times$ SR

(c) Multi-scale results of MS-LapSRN

Figure 7: Quantitative and qualitative results of MS-LapSRN

2.5 Magnification-Arbitrary Network (MetaSR [7])

References

[1] J. Shen, Y. Wang, and J. Zhang, "Asdn: A deep convolutional network for arbitrary scale image super-resolution," *ArXiv*, vol. abs/2010.02414, 2020.

[2] Y. Wang, J. Shen, and J. Zhang, "Deep bi-dense networks for image super-resolution," *CoRR*, vol. abs/1810.04873, 2018.

[3] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," *CoRR*, vol. abs/1807.02758, 2018.

[4] Y. Hu, J. Li, Y. Huang, and X. Gao, "Channel-wise and spatial feature modulation network for single image super-resolution," *CoRR*, vol. abs/1809.11130, 2018.

[5] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," *CoRR*, vol. abs/1704.03915, 2017.

[6] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *CoRR*, vol. abs/1710.01992, 2017.

[7] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, "Meta-sr: A magnification-arbitrary network for super-resolution," *CoRR*, vol. abs/1903.00875, 2019.

[8] Hong Chang, Dit-Yan Yeung, and Yimin Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, pp. I–I, 2004.

[9] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *CoRR*, vol. abs/1501.00092, 2015.

[10] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *CoRR*, vol. abs/1812.10477, 2018.

[11] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," *CoRR*, vol. abs/1511.04491, 2015.

[12] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2790–2798, 2017.

[13] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *CoRR*, vol. abs/1603.05027, 2016.