CS270-B Advanced Digital Image Processing

Lecture 6 Image Super Resolution

(Deep Learning Methods – Fully Supervised ideas)

Yuyao Zhang PhD

zhangyy8@shanghaitech.edu.cn

SIST Building-3 420

Image Super-Resolution

- Low resolution images can be modeled from high resolution images using the below formula, where $\mathbf{D}$ is the degradation function, $\mathbf{I}_y$ is the high resolution image, $\mathbf{I}_x$ is the low resolution image, and $\boldsymbol{\sigma}$ is the noise.

$$\mathrm{I}_x = \mathrm{D}(\mathrm{I}_y; \sigma)$$

- The degradation parameters $\mathbf{D}$ and $\boldsymbol{\sigma}$ are unknown; only the high resolution image and the corresponding low resolution image are provided. The task of the neural network is to find the inverse function of degradation using just the HR and LR image data.

Outline

- **Interpolation methods**
- **Reconstruction based methods**
- **Deep learning based methods**
    - Pre-Upsampling Super Resolution
    - Post-Upsampling Super Resolution
    - Residual Networks
    - Multi-Stage Residual Networks
    - Recursive Networks
    - Progressive Reconstruction Networks
    - Multi-Branch Networks
    - Attention-Based Networks
    - Generative Models

Presampling：
SRCNN & VDSR

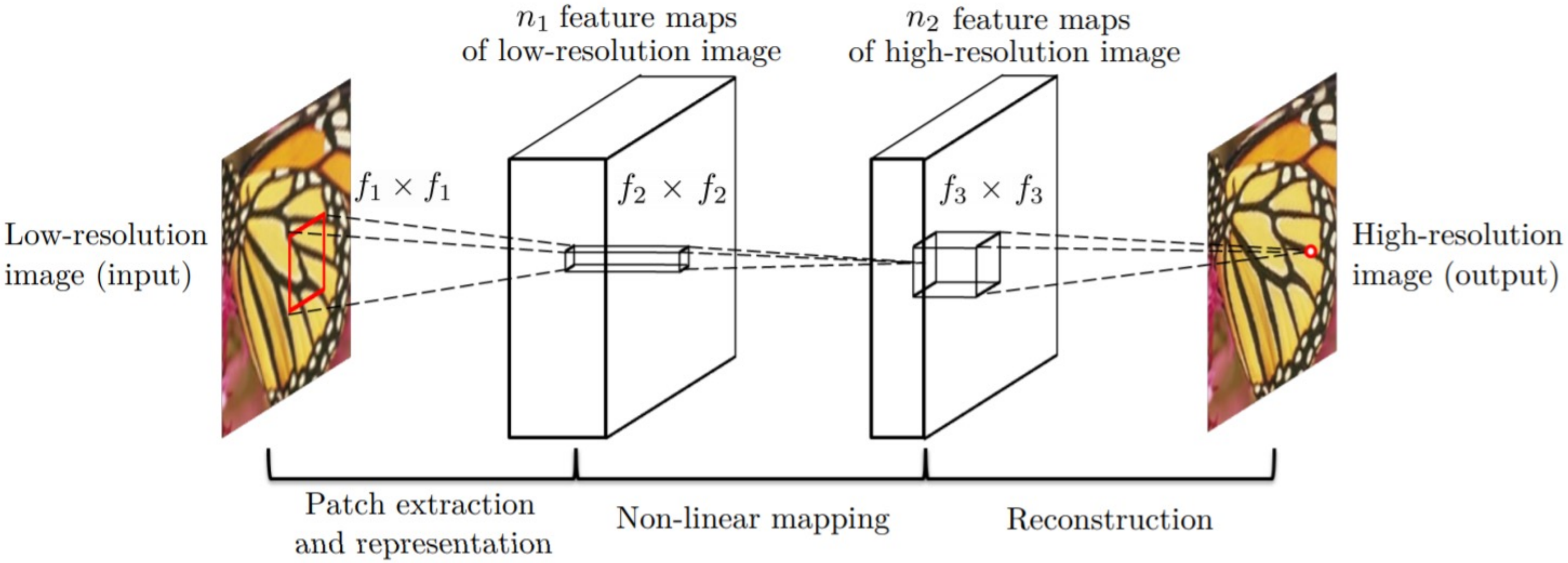
[1] Learning a Deep Convolutional Network for Image Super-Resolution (ECCV 2014)

[2] Image Super-Resolution Using Deep Convolutional Networks (TPAMI 2015)

Pre-Upsampling Super Resolution

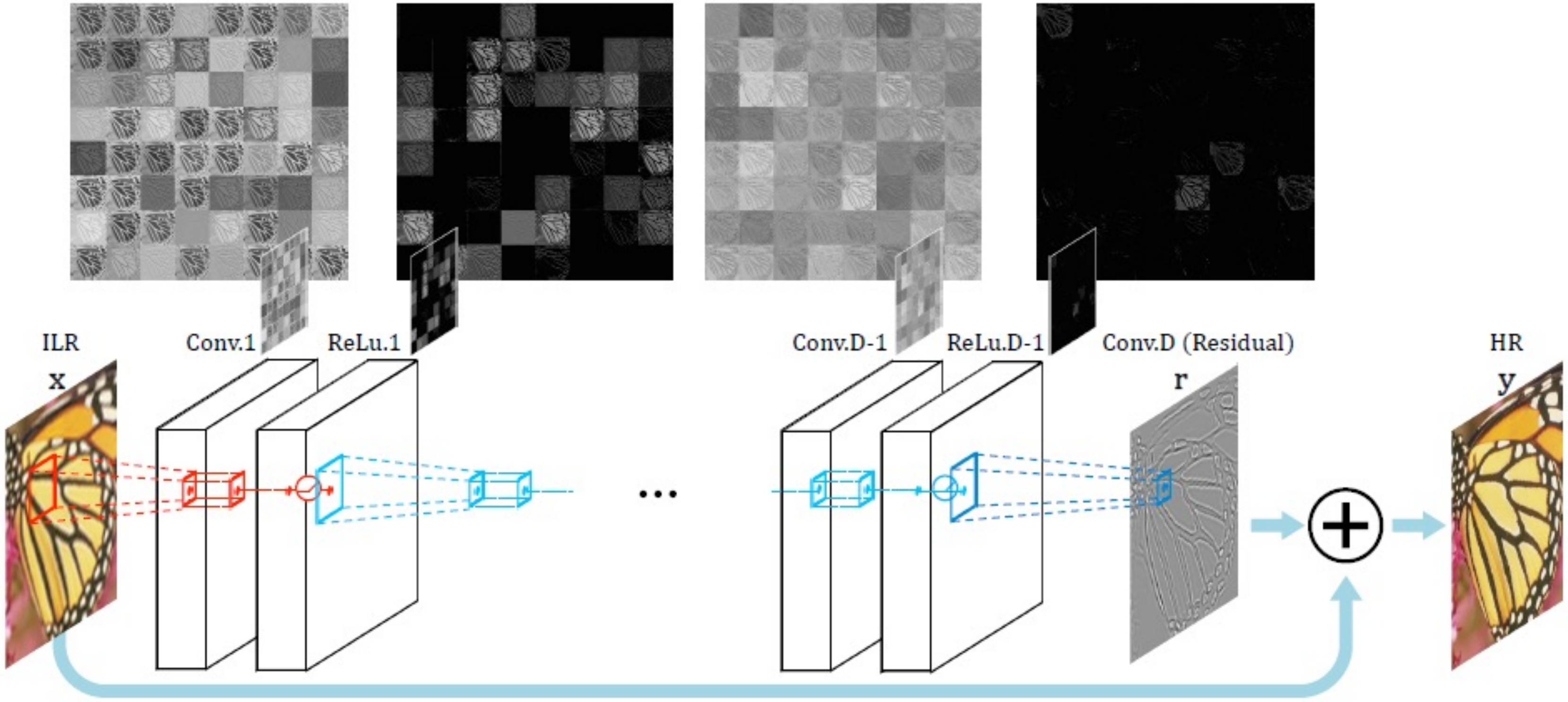
- The methods under this bracket use traditional techniques–like bicubic interpolation and deep learning–to refine an upsampled image.
- The most popular method, SRCNN, was also the first to use deep learning, and has achieved impressive results.

Image Super-Resolution Using Deep Convolutional Networks（SRCNN）



- SRCNN is a simple CNN architecture consisting of three layers:
  - Patch extraction
  - Non-linear mapping
  - Reconstruction

Very Deep Super Resolution（VDSR）



- Very Deep Super Resolution (VDSR) is an improvement on SRCNN

Post-Upsampling Super Resolution：FSRCNN & ESPCN

[1] Accelerating the Super-Resolution Convolutional Neural Networks (ECCV 2016)

[2] Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network (CVPR2016)

FSRCNN

SRCNN (9-1-5)

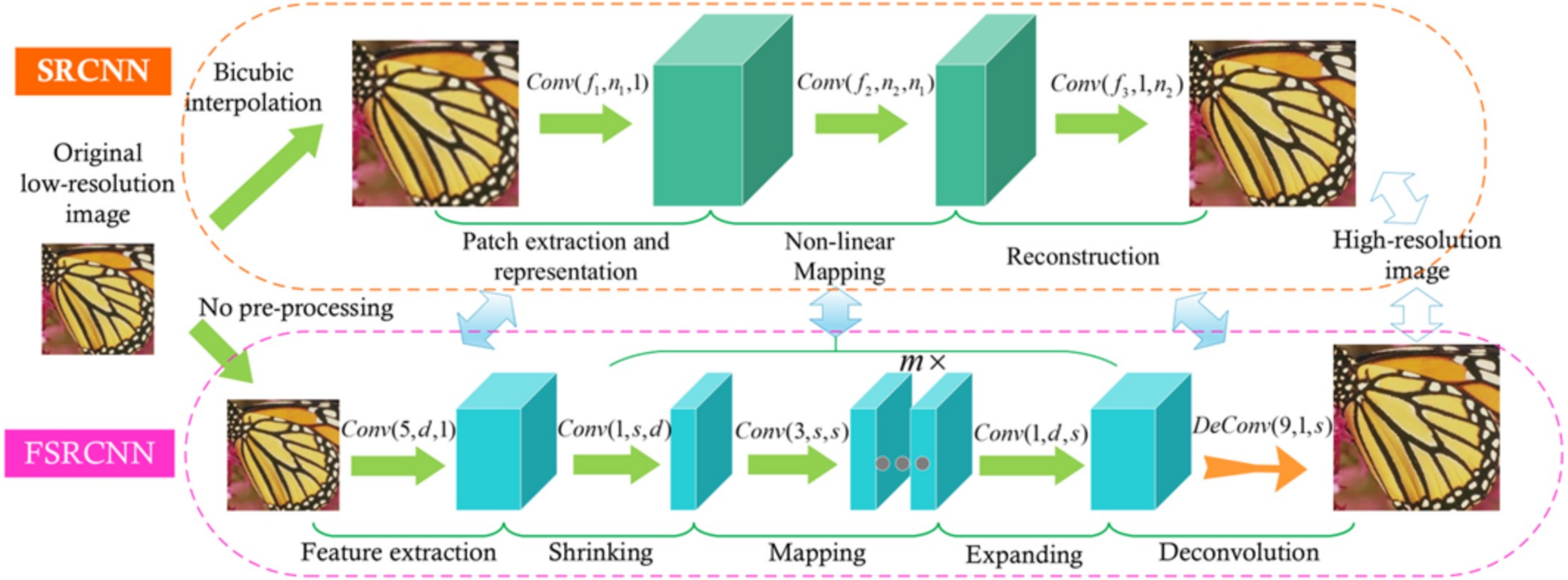
9x9

64 Feature Maps

1x1

32 Feature Maps

5x5



FSRCNN (d,s,m)

5x5

d Feature Maps

1x1

s Feature Maps

3x3

s Feature Maps

1x1

d Feature Maps

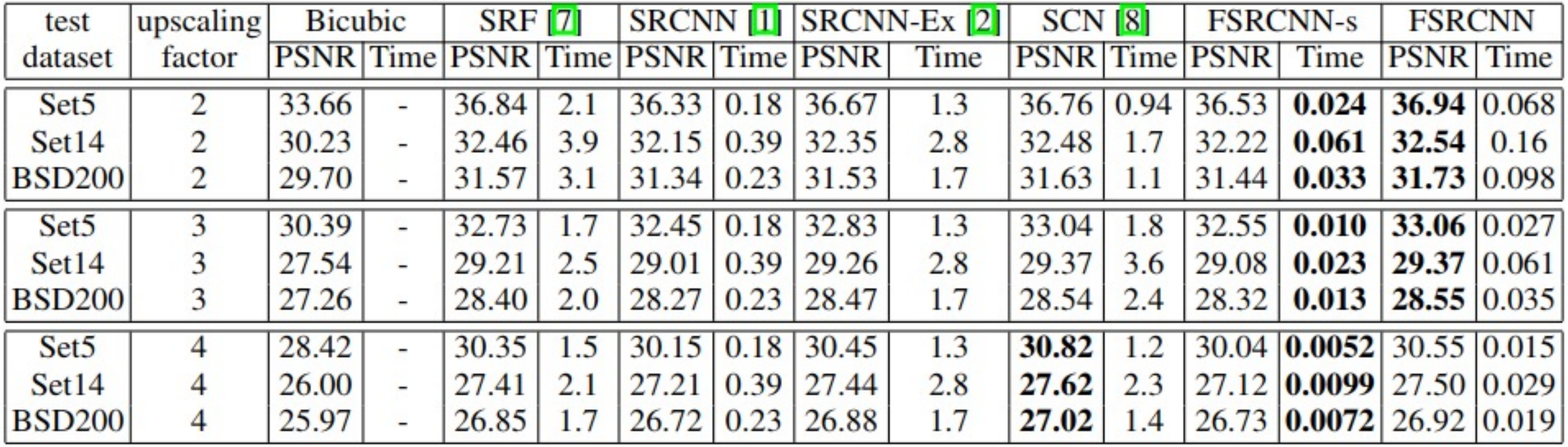
9x9

m times

FSRCNN

- The major changes between SRCNN and FSRCNN are:
  - There is no pre-processing or upsampling at the beginning. The feature extraction took place in the low resolution space.
  - A 1×1 convolution is used after the initial 5×5 convolution to reduce the number of channels, and hence lesser computation and memory, similar to how the Inception network is developed.
  - Multiple 3×3 convolutions are used, instead of having a big convolutional filter, similar to how the VGG network works by simplifying the architecture to reduce the number of parameters.
  - Upsampling is done by using a learned deconvolutional filter, thus improving the model.

Table 3. The results of PSNR (dB) and test time (sec) on three test datasets. All models are trained on the 91-image dataset.



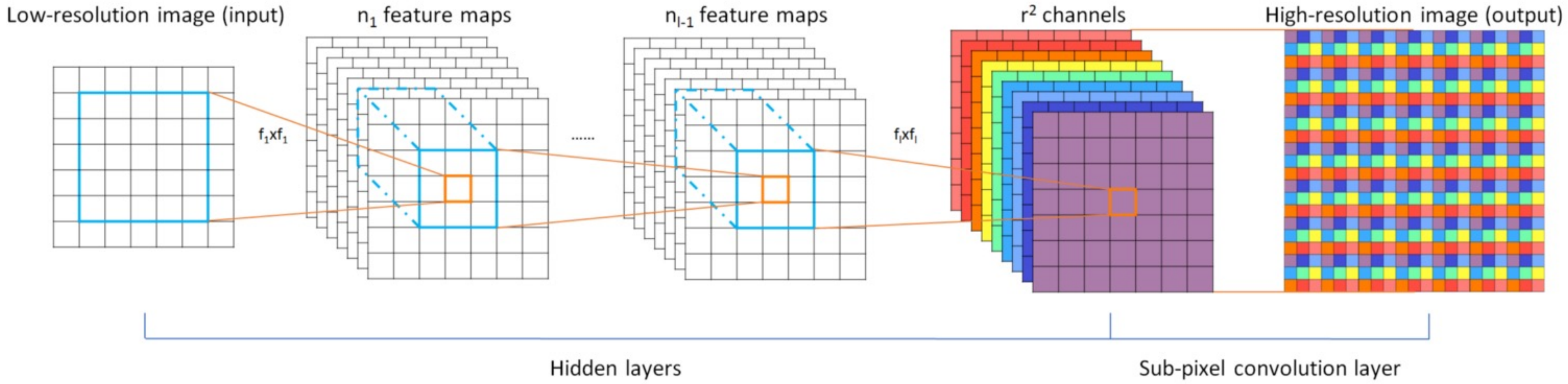
test dataset	upscaling factor	Bicubic		SRF [7]		SRCNN [1]		SRCNN-Ex [2]		SCN [8]		FSRCNN-s		FSRCNN	
		PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
Set5	2	33.66	-	36.84	2.1	36.33	0.18	36.67	1.3	36.76	0.94	36.53	**0.024**	**36.94**	0.068
Set14	2	30.23	-	32.46	3.9	32.15	0.39	32.35	2.8	32.48	1.7	32.22	**0.061**	**32.54**	0.16
BSD200	2	29.70	-	31.57	3.1	31.34	0.23	31.53	1.7	31.63	1.1	31.44	**0.033**	**31.73**	0.098
Set5	3	30.39	-	32.73	1.7	32.45	0.18	32.83	1.3	33.04	1.8	32.55	**0.010**	**33.06**	0.027
Set14	3	27.54	-	29.21	2.5	29.01	0.39	29.26	2.8	29.37	3.6	29.08	**0.023**	**29.37**	0.061
BSD200	3	27.26	-	28.40	2.0	28.27	0.23	28.47	1.7	28.54	2.4	28.32	**0.013**	**28.55**	0.035
Set5	4	28.42	-	30.35	1.5	30.15	0.18	30.45	1.3	**30.82**	1.2	30.04	**0.0052**	30.55	0.015
Set14	4	26.00	-	27.41	2.1	27.21	0.39	27.44	2.8	**27.62**	2.3	27.12	**0.0099**	27.50	0.029
BSD200	4	25.97	-	26.85	1.7	26.72	0.23	26.88	1.7	**27.02**	1.4	26.73	**0.0072**	26.92	0.019

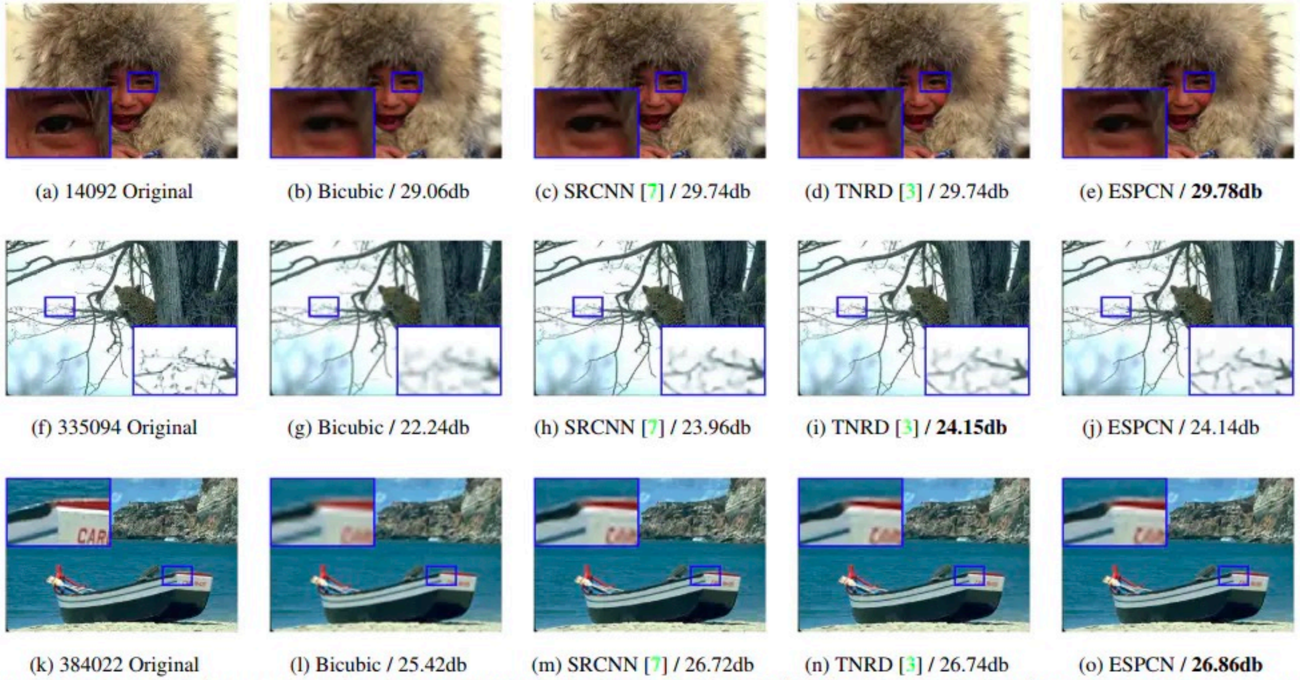
ESPCN

- ESPCN introduces the concept of sub-pixel convolution to replace the deconvolutional layer for upsampling. This solves two problems associated with it:
  - Deconvolution happens in the high resolution space, and thus is more computationally expensive.
  - It resolves the checkerboard issue in deconvolution, which occurs due to the overlap operation of convolution (shown below).

ESPCN

- Sub-pixel convolution works by converting depth to space, as seen in the figure below. Pixels from multiple channels in a low resolution image are rearranged to a single channel in a high resolution image. To give an example, an input image of size 5×5×4 can rearrange the pixels in the final four channels to a single channel, resulting in a 10×10 HR image.



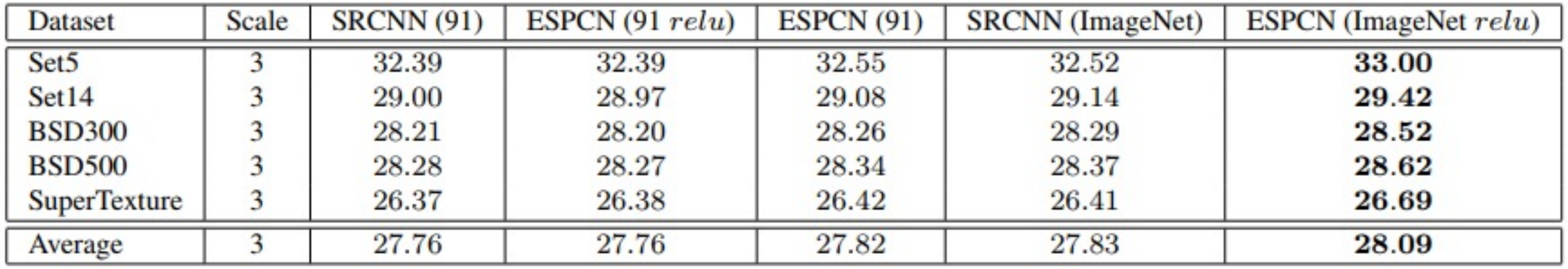


(a) 14092 Original (b) Bicubic / 29.06db (c) SRCNN [7] / 29.74db (d) TNRD [3] / 29.74db (e) ESPCN / **29.78db**

(f) 335094 Original (g) Bicubic / 22.24db (h) SRCNN [7] / 23.96db (i) TNRD [3] / **24.15db** (j) ESPCN / 24.14db

(k) 384022 Original (l) Bicubic / 25.42db (m) SRCNN [7] / 26.72db (n) TNRD [3] / 26.74db (o) ESPCN / **26.86db**

Figure 6. Super-resolution examples for ”14092”, ”335094” and ”384022” from **BSD500** with an upscaling factor of 3. PSNR values are shown under each sub-figure.



Dataset	Scale	SRCNN (91)	ESPCN (91 $relu$)	ESPCN (91)	SRCNN (ImageNet)	ESPCN (ImageNet $relu$)
Set5	3	32.39	32.39	32.55	32.52	**33.00**
Set14	3	29.00	28.97	29.08	29.14	**29.42**
BSD300	3	28.21	28.20	28.26	28.29	**28.52**
BSD500	3	28.28	28.27	28.34	28.37	**28.62**
SuperTexture	3	26.37	26.38	26.42	26.41	**26.69**
Average	3	27.76	27.76	27.82	27.83	**28.09**

Table 1. The mean PSNR (dB) for different models. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and other methods (p-value < 0.001 with paired t-test).

Post-Upsampling Super-Resolution

- Since the feature extraction process in pre-upsampling SR occurs in the high resolution space, the computational power required is also on the higher end. Post-upsampling SR tries to solve this by doing feature extraction in the lower resolution space, then doing upsampling only at the end, therefore significantly reducing computation.
- Also, instead of using simple bicubic interpolation for upsampling, a learned upsampling in the form of deconvolution/sub-pixel convolution is used, thus making the network trainable end-to-end.

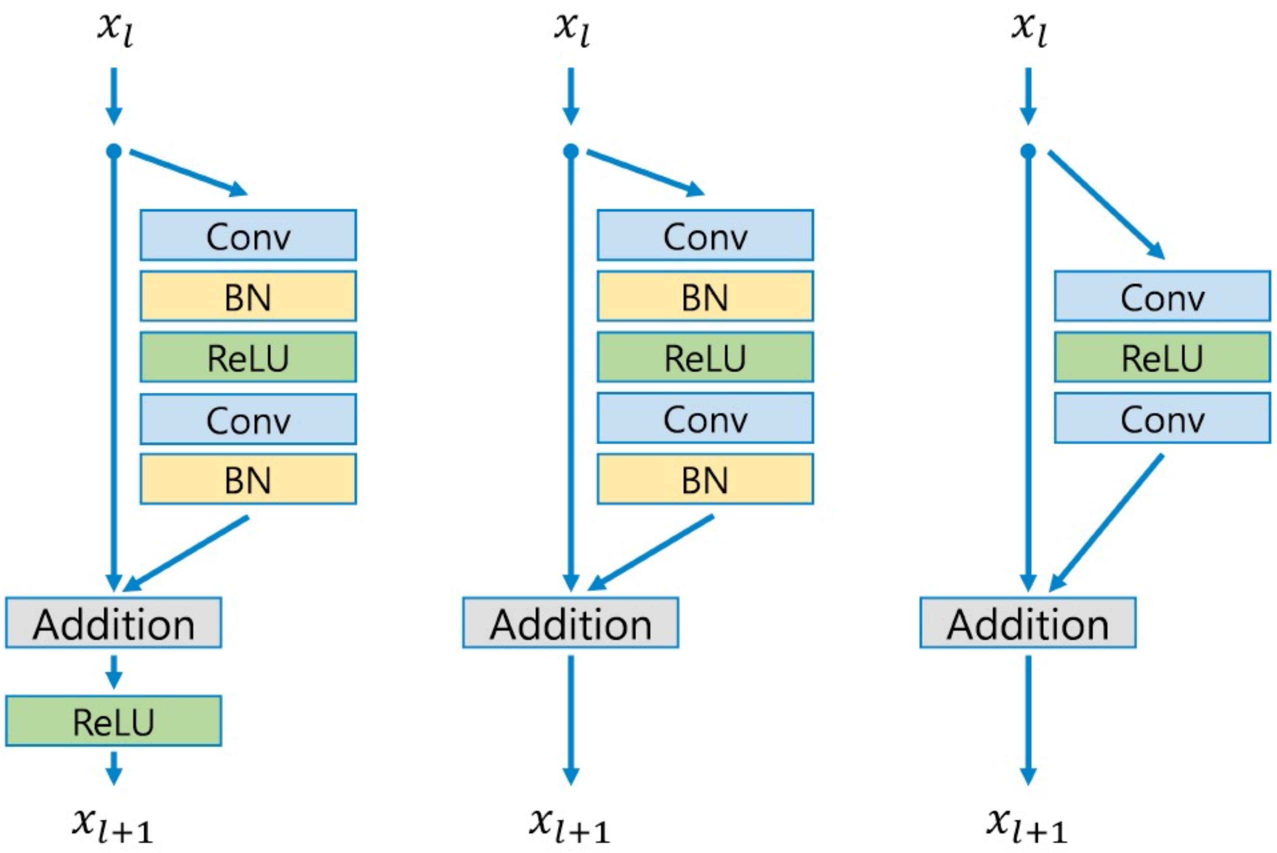
Residual Networks：EDSR & MDSR & CARN

[1] EDSR & MDSR Enhanced Deep Residual Networks for Single Image Super-Resolution (CVPR 2017)

[2] CARN: Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network (ECCV 2018)

EDSR

- The EDSR architecture is based on the SRResNet architecture, consisting of multiple residual blocks.
- The major difference from SRResNet is that the Batch Normalization layers are removed. The author states that BN normalizes the input, thus limiting the range of the network; removal of BN results in an improvement in accuracy.



(a) Original (b) SRResNet (c) Proposed

EDSR

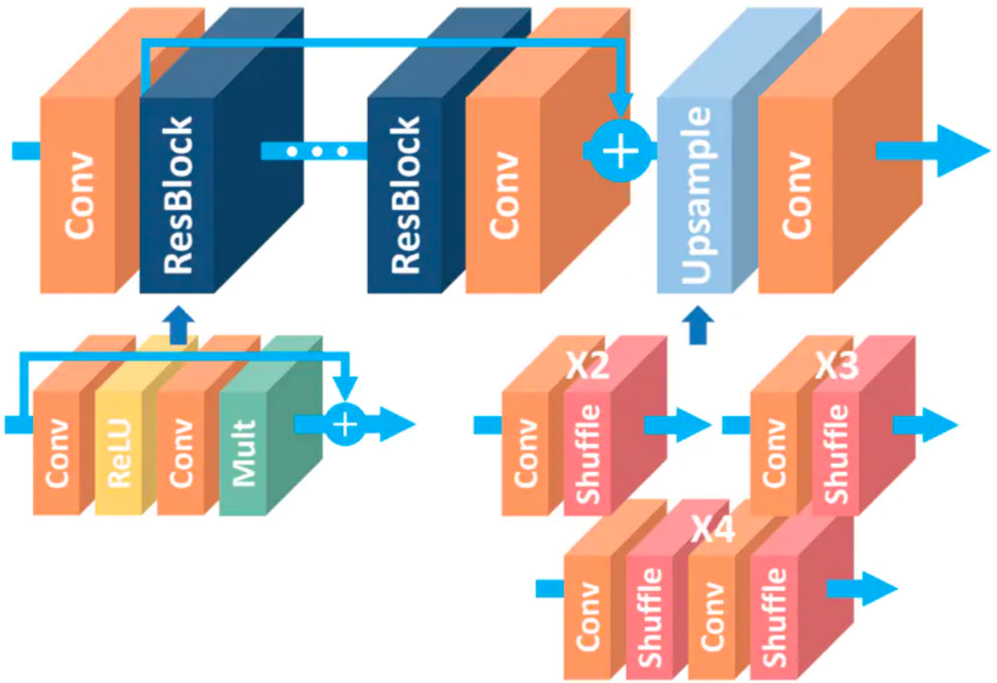


Figure 3: The architecture of the proposed single-scale SR network (EDSR).

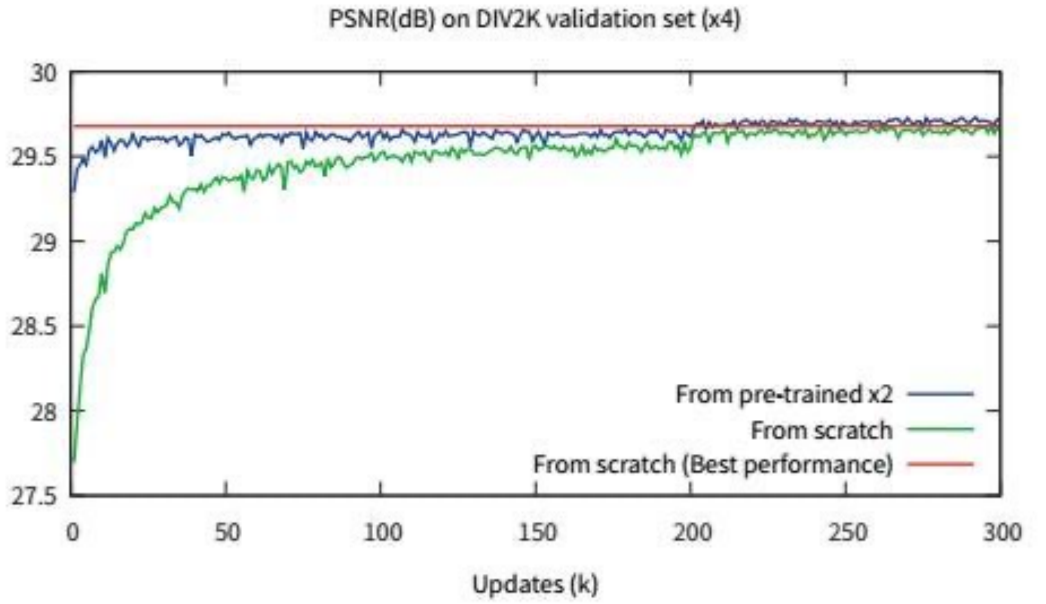
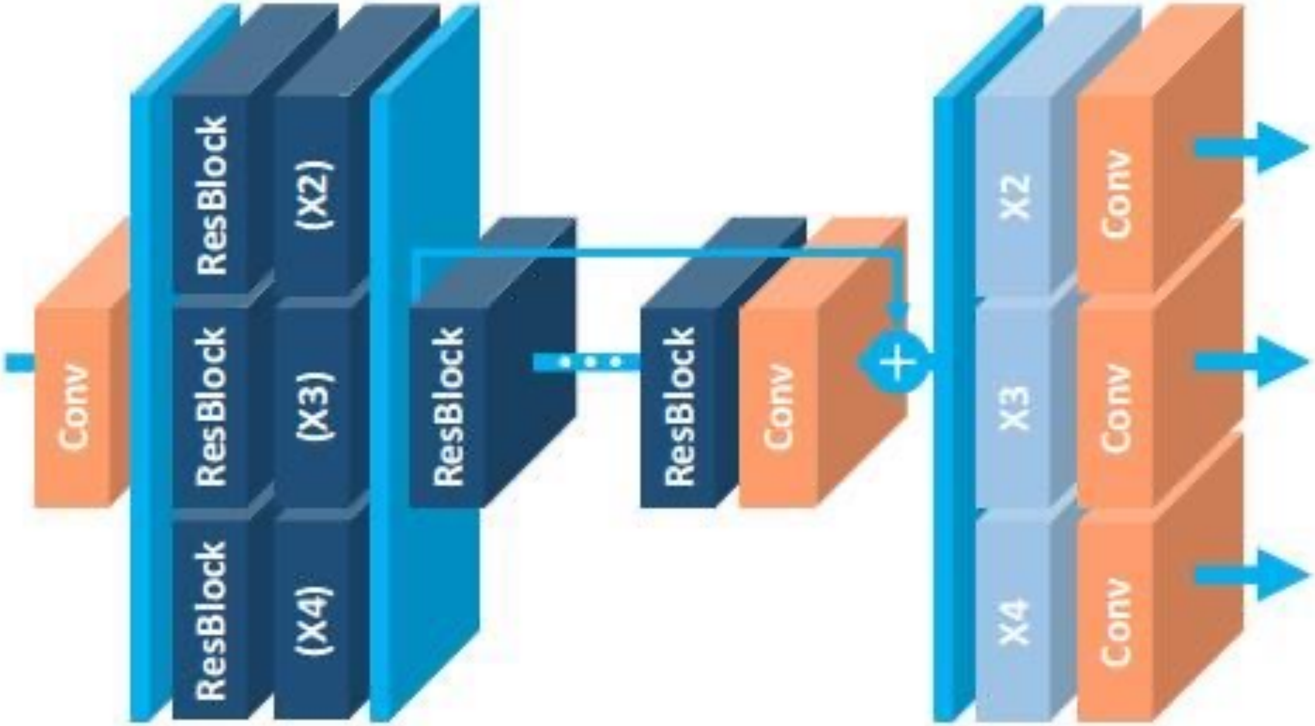


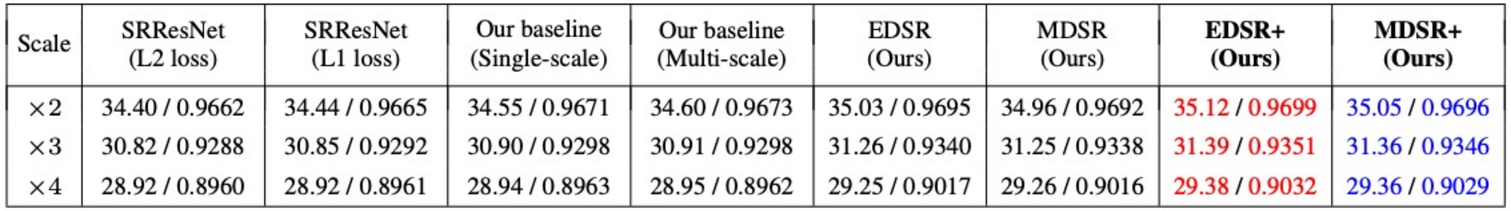
Figure 4: Effect of using pre-trained ×2 network for ×4 model (EDSR). The red line indicates the best performance of green line. 10 images are used for validation during training.

MDSR

- MDSR is an extension of EDSR, with multiple input and output modules that give corresponding resolution outputs at 2x, 3x, and 4x.
- At the beginning, the pre-processing modules for scale-specific input are present consisting of two residual blocks with 5×5 kernels. A large kernel is used for the pre-processing layers to keep the network shallow, while still achieving a high receptive field.
- At the end of the scale-specific pre-processing modules are the shared residual blocks, which is a common block for data of all resolutions. Finally, after the shared residual blocks are the scale-specific upsampling modules.

MDSR



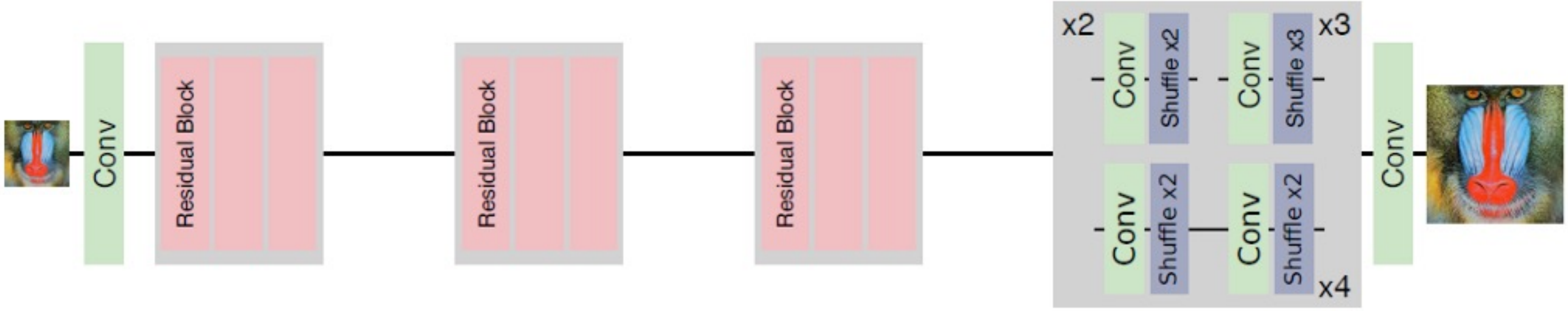


Scale	SRResNet (L2 loss)	SRResNet (L1 loss)	Our baseline (Single-scale)	Our baseline (Multi-scale)	EDSR (Ours)	MDSR (Ours)	**EDSR+** **(Ours)**	**MDSR+** **(Ours)**
×2	34.40 / 0.9662	34.44 / 0.9665	34.55 / 0.9671	34.60 / 0.9673	35.03 / 0.9695	34.96 / 0.9692	35.12 / 0.9699	35.05 / 0.9696
×3	30.82 / 0.9288	30.85 / 0.9292	30.90 / 0.9298	30.91 / 0.9298	31.26 / 0.9340	31.25 / 0.9338	31.39 / 0.9351	31.36 / 0.9346
×4	28.92 / 0.8960	28.92 / 0.8961	28.94 / 0.8963	28.95 / 0.8962	29.25 / 0.9017	29.26 / 0.9016	29.38 / 0.9032	29.36 / 0.9029

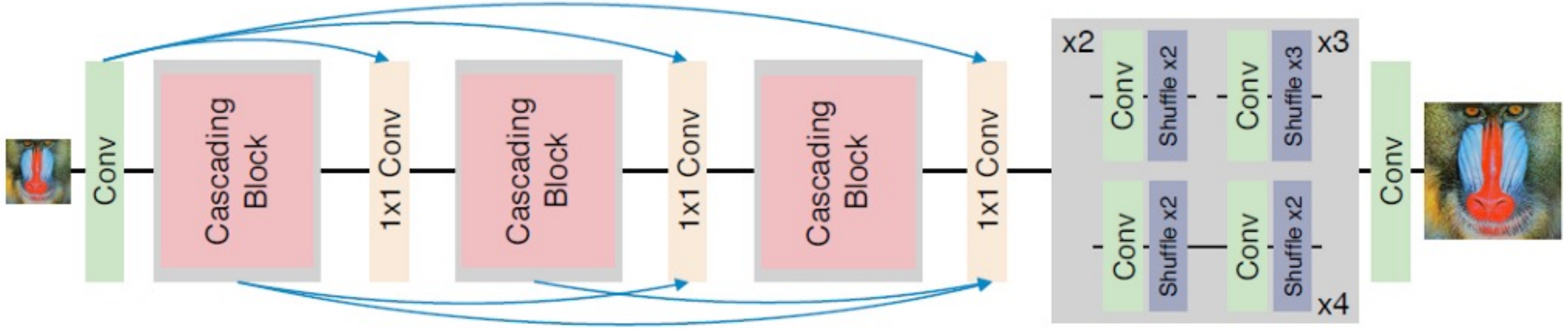
CARN

- In the paper, the authors have proposed the following advancements on top of a traditional residual network:
- A cascading mechanism at both the local and global level, to incorporate features from multiple layers and give the network the ability to receive more information.
- In addition to CARN, a smaller CARN-M is proposed to have a lighter architecture, without much deterioration in results, with the help of recursive network architecture.

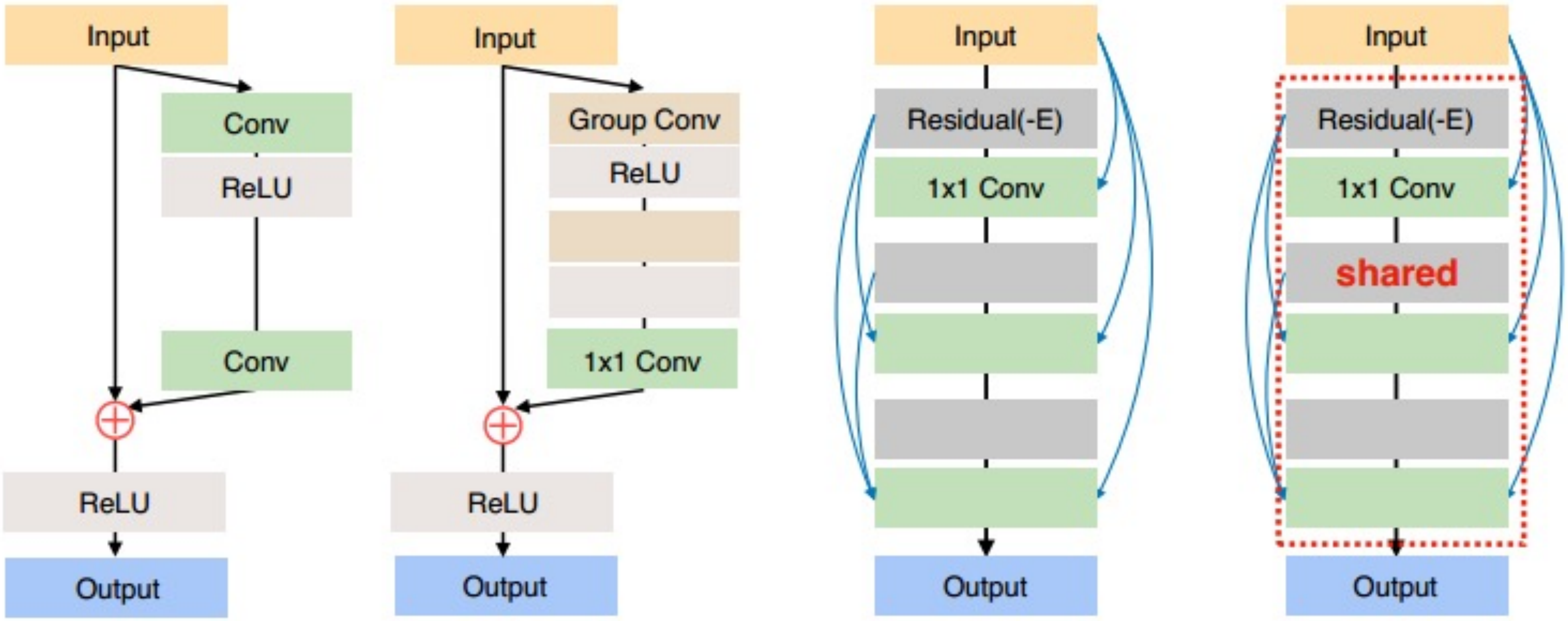
CARN



(a) Plain ResNet



(b) Cascading Residual Network (CARN)



(a) Residual Block (b) Residual-E Block (c) Cascading Block (d) Recursive Block

Fig. 3: Simplified structures of (a) residual block (b) efficient residual block (residual-E), (c) cascading block and (d) recursive cascading block. The ⊕ operations in (a) and (b) are element-wise addition for residual learning.

Multi-Stage Residual Networks：BTSRN

[1] EDSR & MDSR Balance two-stage Residual Networks for Image Super-Resolution (CVPR-W 2017)

Multi-Stage Residual Networks

- To deal with the task of feature extraction separately in the low-resolution space and high-resolution space, a multi-stage design is considered in a few architectures to improve their performance. The first stage predicts the coarse features, while the later stage improves on it. Let's discuss an architecture involving one of these multi-stage networks.

BTSRN

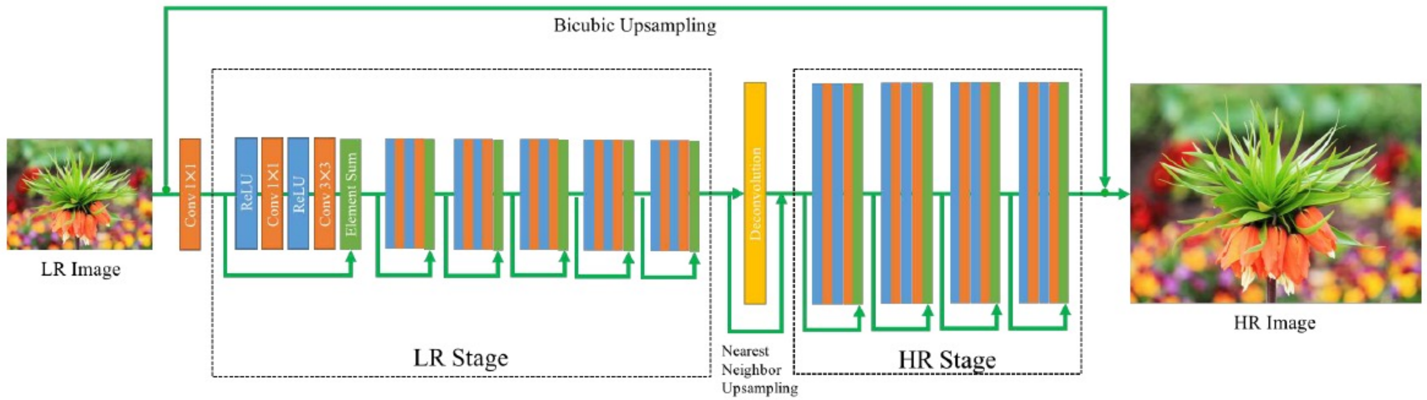
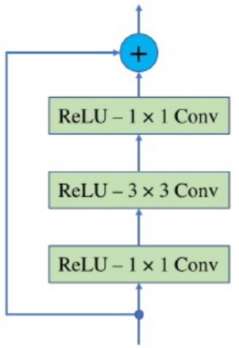
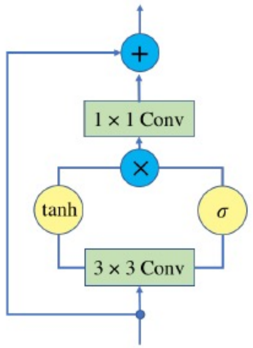


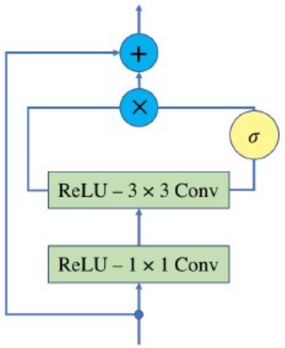
Figure 1: Architecture of the proposed network



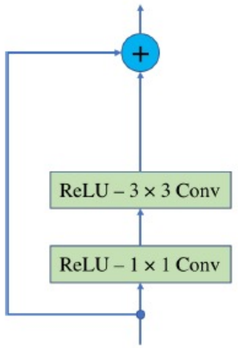
(a) Residual block in PixelCNN



(b) Gated conv in PixelCNN



(c) Gated conv in PixelCNN++



(d) PConv block in proposed BTSRN model

BTSRN

- BTSRN consists of two stages: a low resolution (LR) stage and a high resolution (HR) stage.
- The LR stage consists of 6 residual blocks, whereas the HR stage contains 4 residual blocks. Convolution in the HR stage requires more computation than in the LR stage, as the input size is higher.
- The number of blocks in both the stages are determined in such a way as to achieve a trade-off between accuracy and performance.

BTSRN

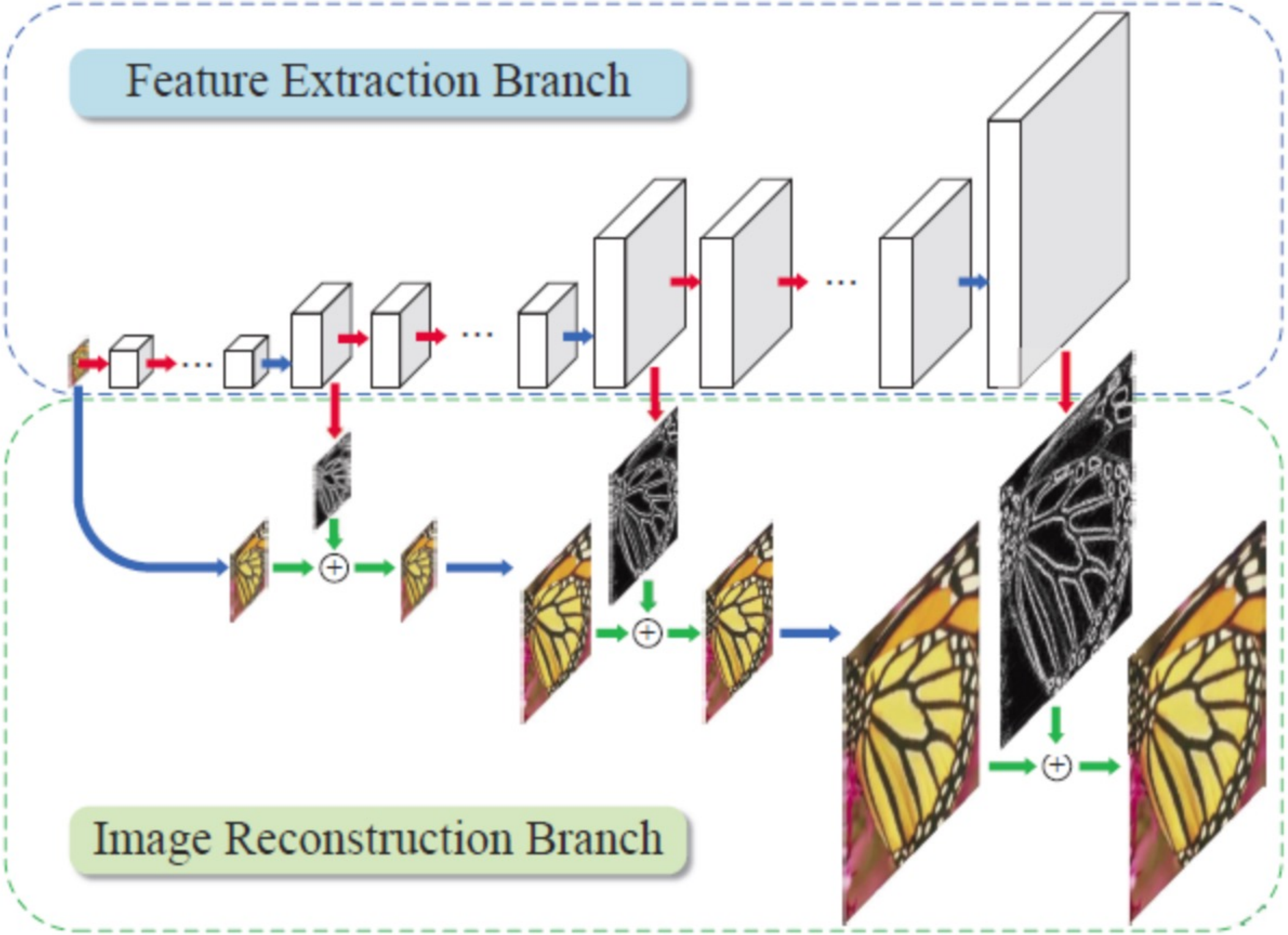
- BTSRN consists of two stages: a low resolution (LR) stage and a high resolution (HR) stage.
- The LR stage consists of 6 residual blocks, whereas the HR stage contains 4 residual blocks. Convolution in the HR stage requires more computation than in the LR stage, as the input size is higher.
- The number of blocks in both the stages are determined in such a way as to achieve a trade-off between accuracy and performance.

Progressive Reconstruction Networks：LAPSRN

[1] EDSR & MDSR Balance two-stage Residual Networks for Image Super-Resolution (CVPR-W 2017)

[2] CARN: Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network (ECCV 2018)

LAPSRN

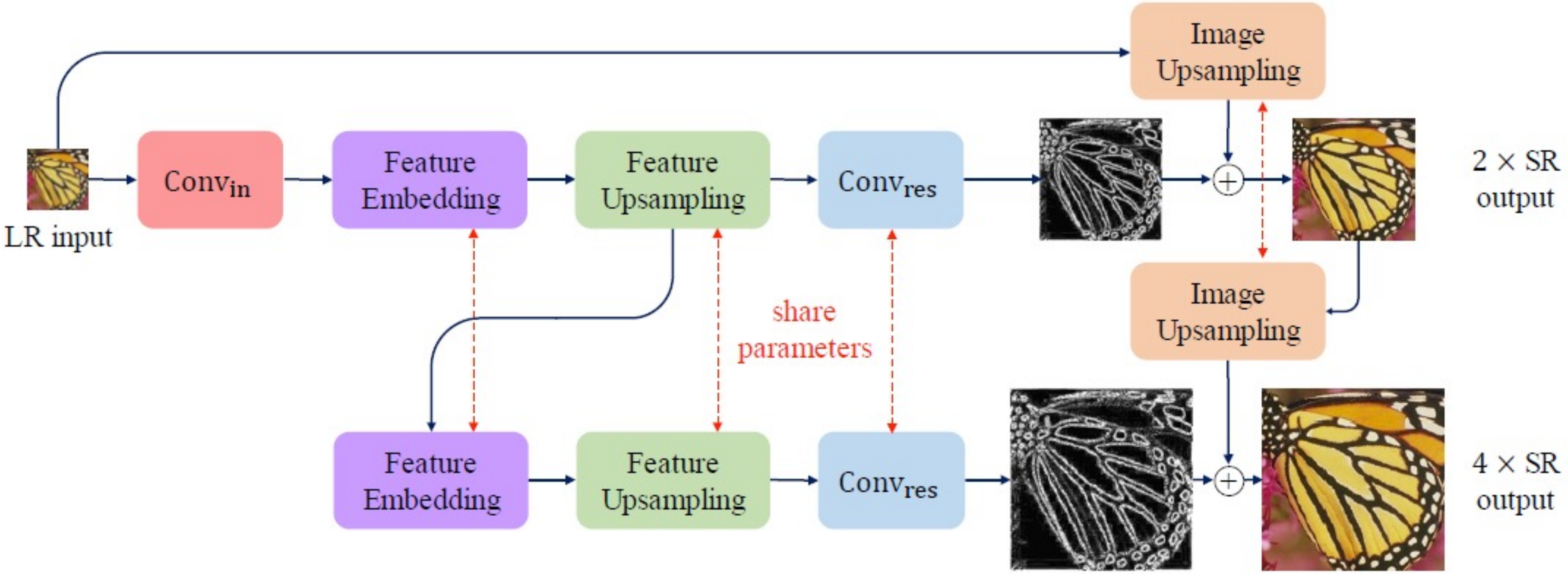


LAPSRN

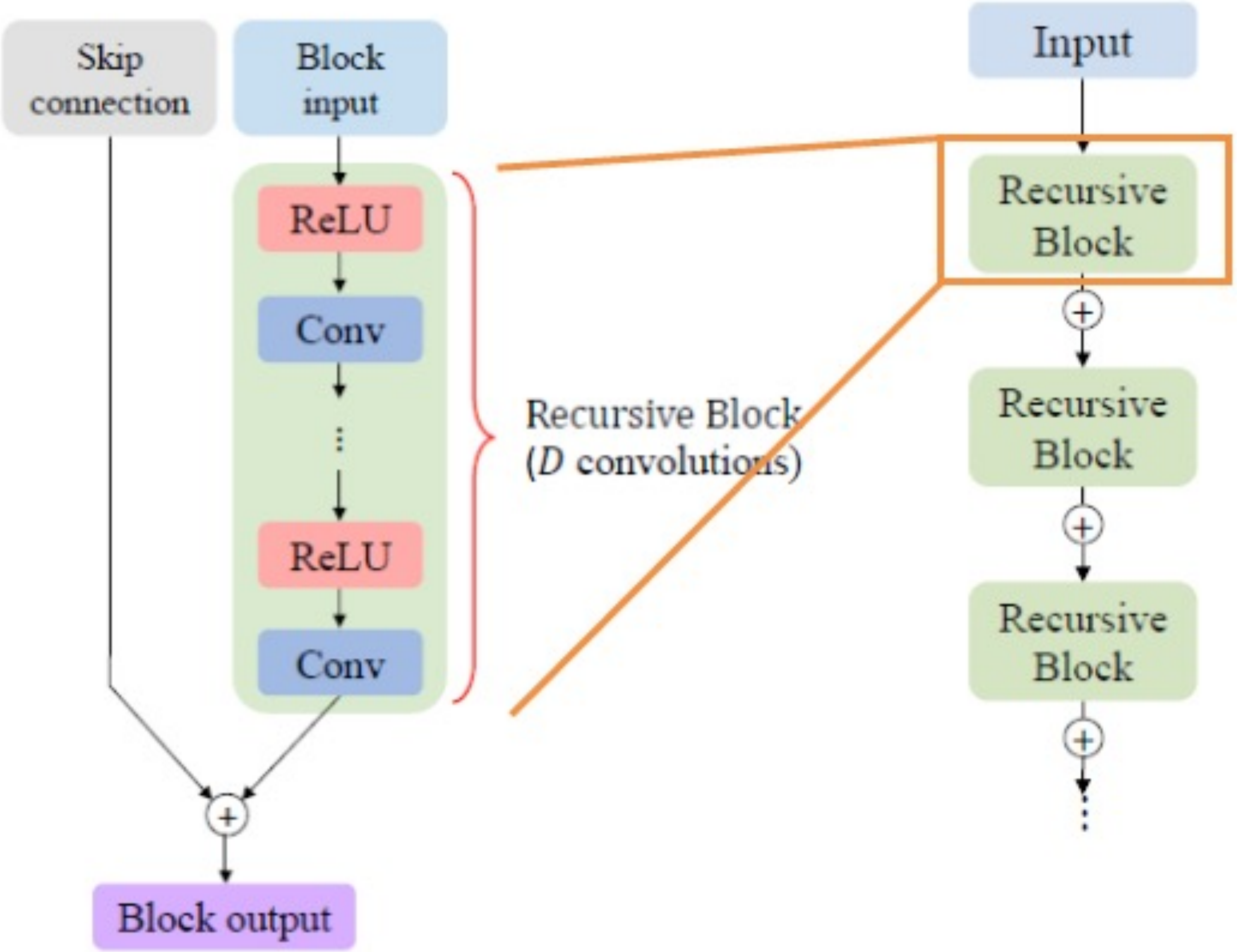
- LAPSRN, or MS-LAPSRN, consists of a Laplacian pyramid structure which can upscale images to 2x, 4x, and 8x using a step-by-step approach.
- LAPSRN consists of multiple stages. The network consists of two branches: the Feature Extraction Branch and the Image Reconstruction Branch.

LAPSRN

- Recursive block



LAPSRN


Skip connection
Block input
ReLU
Conv
ReLU
Conv
Recursive Block
(*D* convolutions)
Block output
Input
Recursive Block
Recursive Block
Recursive Block

Summary

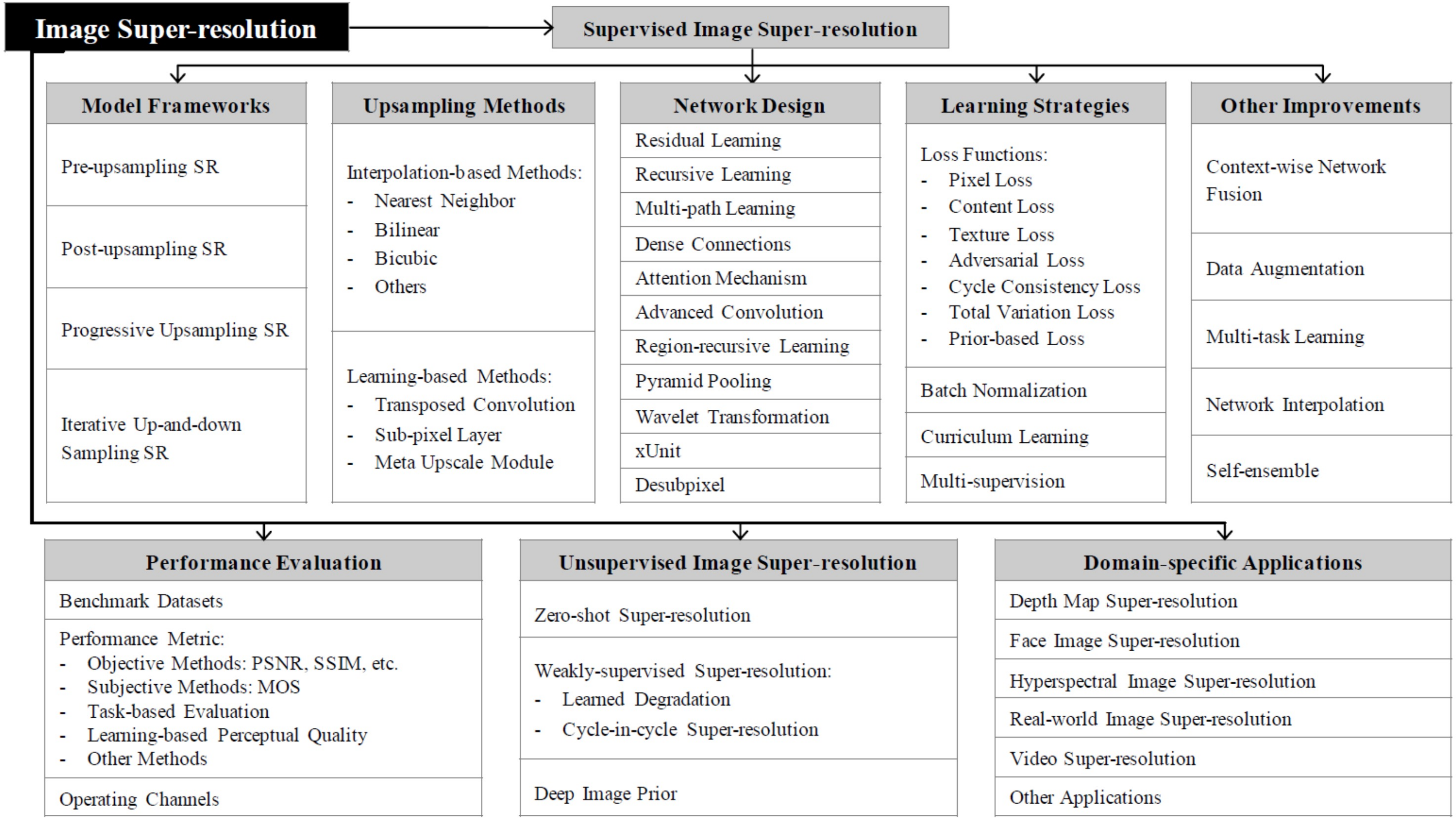


Image Super-resolution

Supervised Image Super-resolution

Model Frameworks

- Pre-upsampling SR
- Post-upsampling SR
- Progressive Upsampling SR
- Iterative Up-and-down Sampling SR

Upsampling Methods

Interpolation-based Methods:
- Nearest Neighbor
- Bilinear
- Bicubic
- Others

Learning-based Methods:
- Transposed Convolution
- Sub-pixel Layer
- Meta Upscale Module

Network Design

- Residual Learning
- Recursive Learning
- Multi-path Learning
- Dense Connections
- Attention Mechanism
- Advanced Convolution
- Region-recursive Learning
- Pyramid Pooling
- Wavelet Transformation
- xUnit
- Desubpixel

Learning Strategies

Loss Functions:
- Pixel Loss
- Content Loss
- Texture Loss
- Adversarial Loss
- Cycle Consistency Loss
- Total Variation Loss
- Prior-based Loss

Batch Normalization

Curriculum Learning

Multi-supervision

Other Improvements

- Context-wise Network Fusion
- Data Augmentation
- Multi-task Learning
- Network Interpolation
- Self-ensemble

Performance Evaluation

Benchmark Datasets

Performance Metric:
- Objective Methods: PSNR, SSIM, etc.
- Subjective Methods: MOS
- Task-based Evaluation
- Learning-based Perceptual Quality
- Other Methods

Operating Channels

Unsupervised Image Super-resolution

Zero-shot Super-resolution

Weakly-supervised Super-resolution:
- Learned Degradation
- Cycle-in-cycle Super-resolution

Deep Image Prior

Domain-specific Applications

- Depth Map Super-resolution
- Face Image Super-resolution
- Hyperspectral Image Super-resolution
- Real-world Image Super-resolution
- Video Super-resolution
- Other Applications