2024-04-16

Lightweight Image Super-Resolution with Enhanced CNN

김도완, 임채연

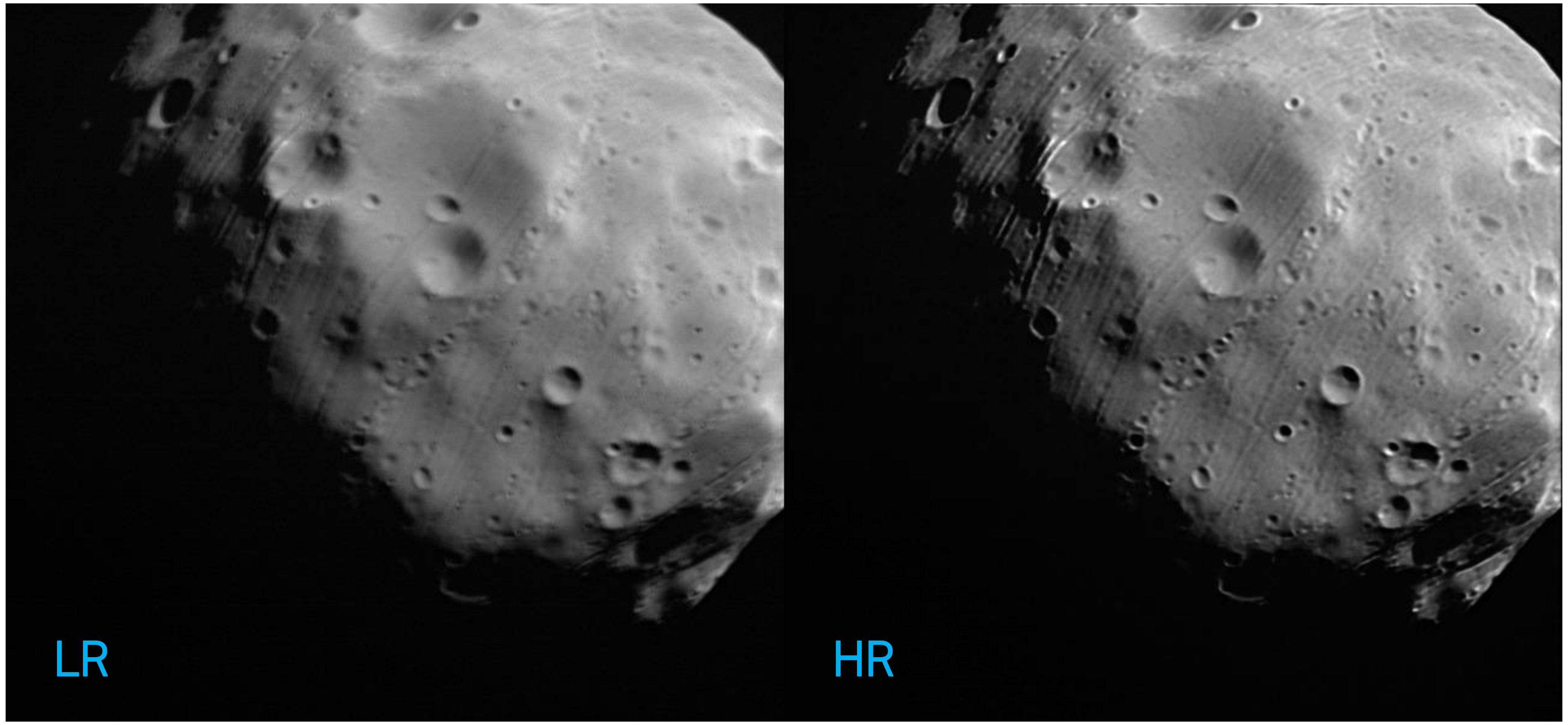
since 1978
DONG SEOUL UNIVERSITY

01

SISR

Single Image Super Resolution

SISR aims at recovering a High-Resolution image from a Low-Resolution

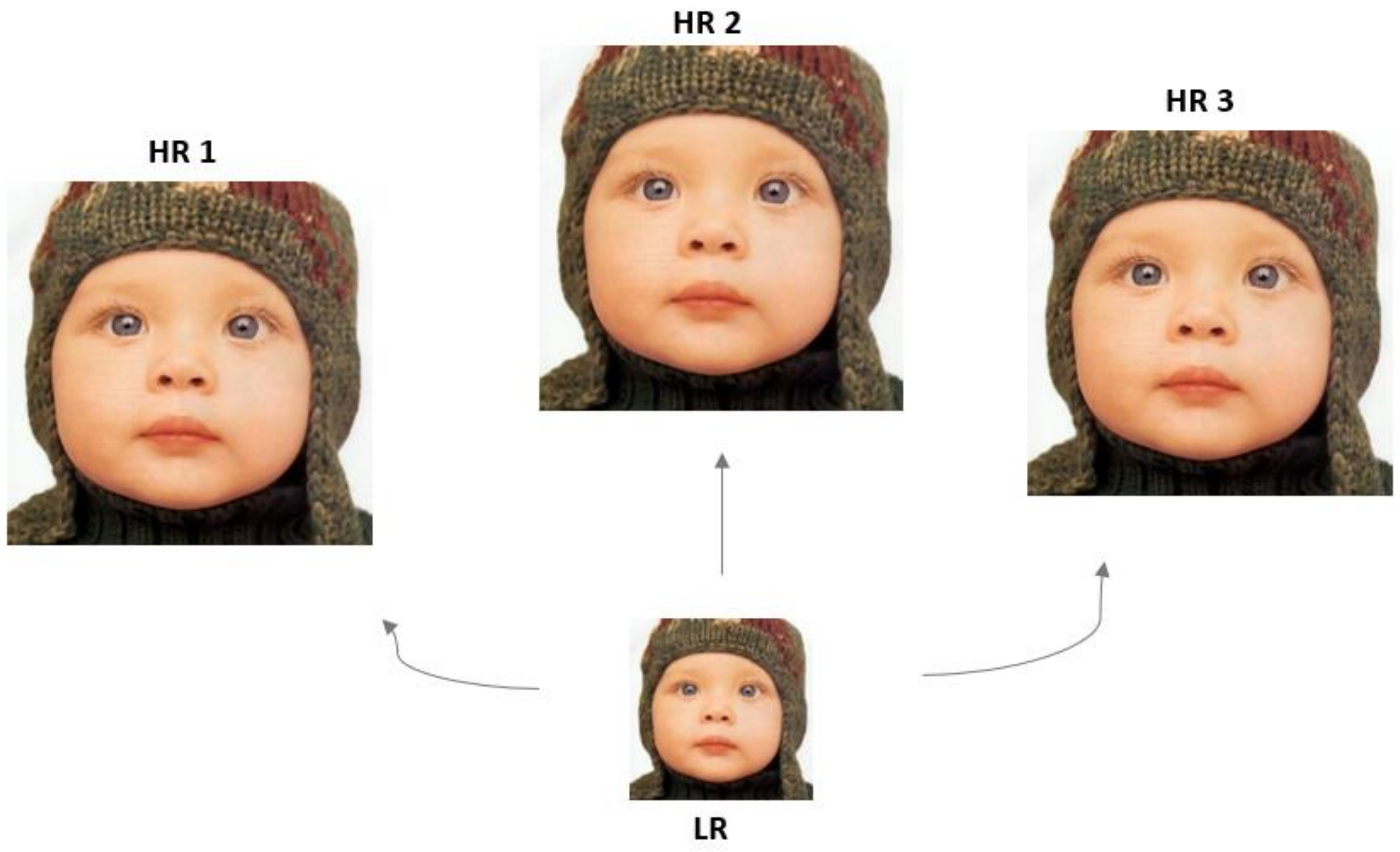


mutiple HR images can be downsampled to the same LR image

김도완, 임채연

ill-posed problem

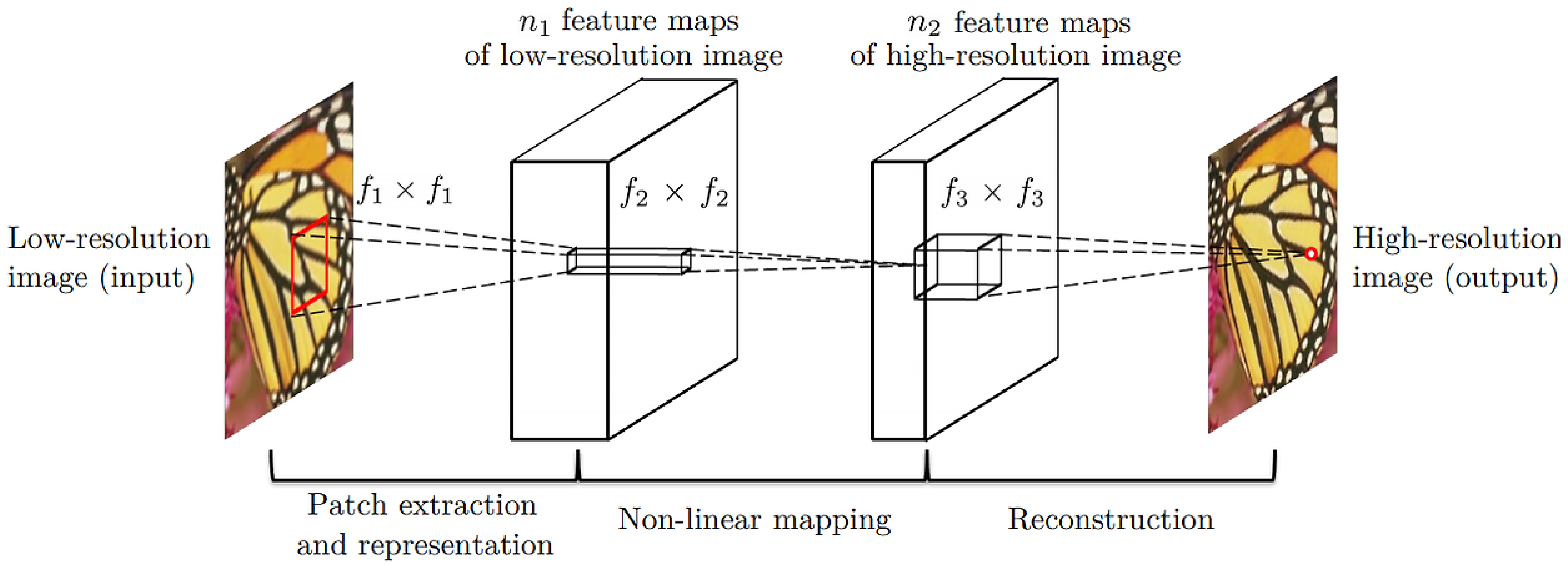
one which doesn't have a unique solution



Prior knowledge methods were developed by constraining the solution space

SRCNN(Super Resolution Convolution Neural Network)

Proposed a pioneering three-layer



Obtain the SR image in a pixel mapping manner

Deep CNNs based cascade structures for SISR

Development of Big Data and GPU → Deep CNN applied in SISR

Deep CNNs based cascade structures for SISR

SR techniques based on Deep CNNs

1. Based in High-Frequency features

→ Higher computational cost & memory consumption

2. Based in Low-Frequency features

→ Ignore detailed High-Frequency features

3. Combination High-Frequency and Low-Frequency

→ High-Quality Image

Deep CNNs based cascade structures for SISR

Combination High-Frequency and Low Frequency Method

Good performance in Cascade structure

Performance

Efficiency

Deep CNNs based cascade structures for SISR

SR: Performance

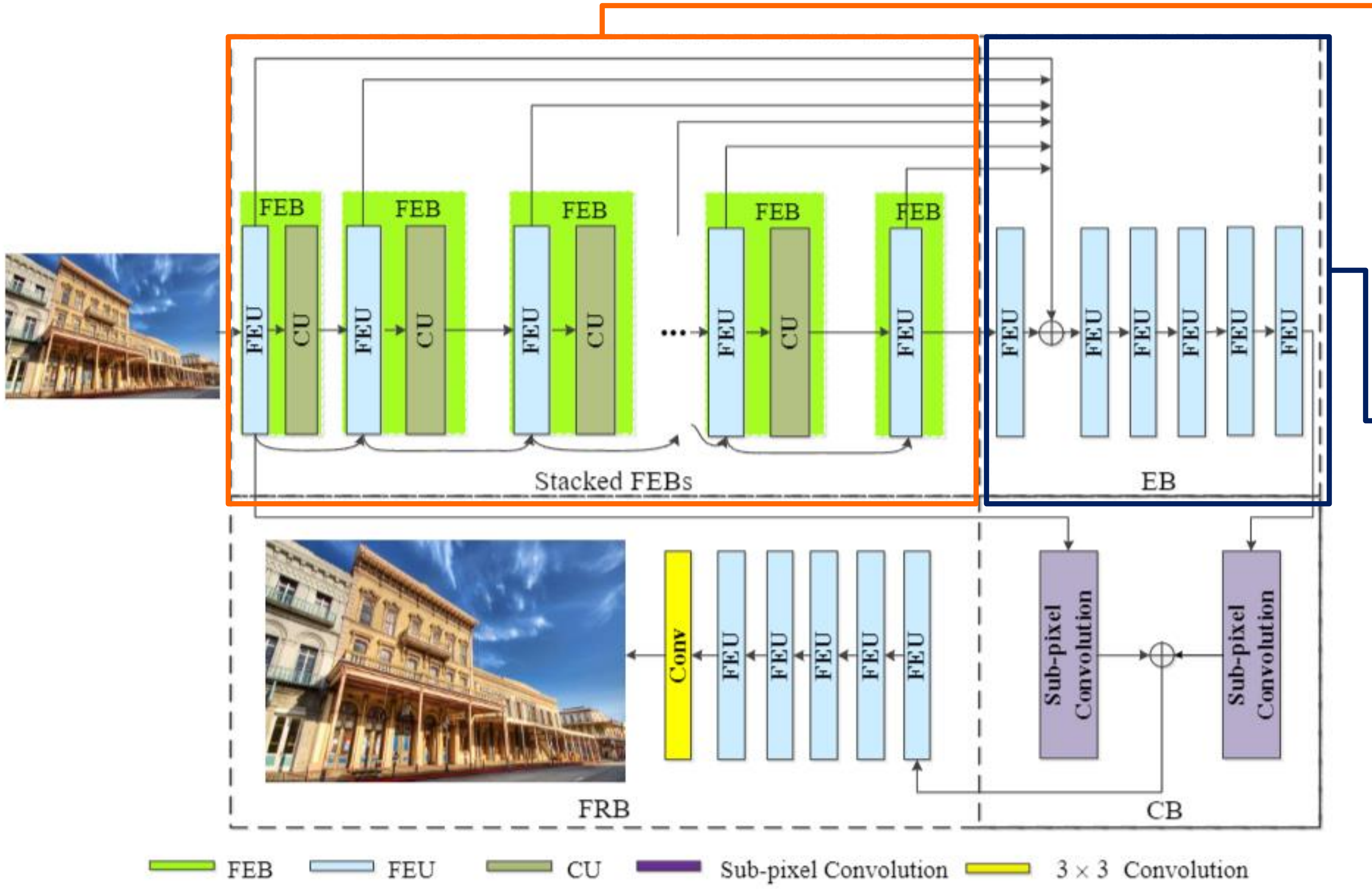
- Coarse-to-fine CNN
- CDN(Cascading Dense Network)

SR: Efficiency

- CARN(Cascading Residual Network)

Deep CNNs based cascade structures for SISR

- Coarse-to-fine CNN [Performance]



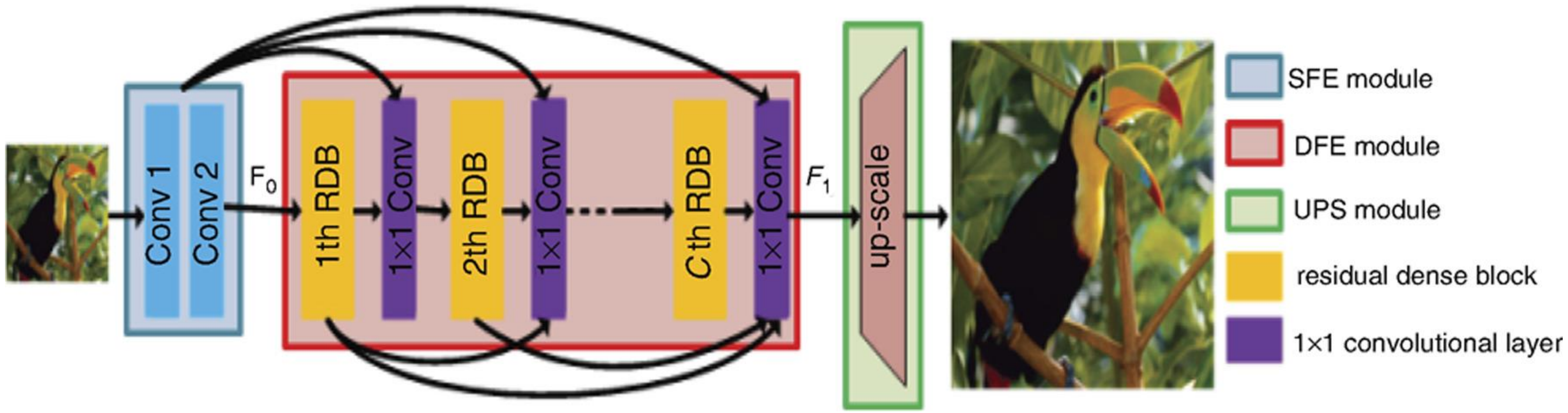
1. Extract Low-Frequency feature
2. Learn High-Frequency feature using Enhancement Block

Deep CNNs based cascade structures for SISR

- CDN(Cascading Dense Network) [Performance]

1. Extract hierarchical features from each convolution layer

2. Residual dense block can eliminate Vanishing Gradient

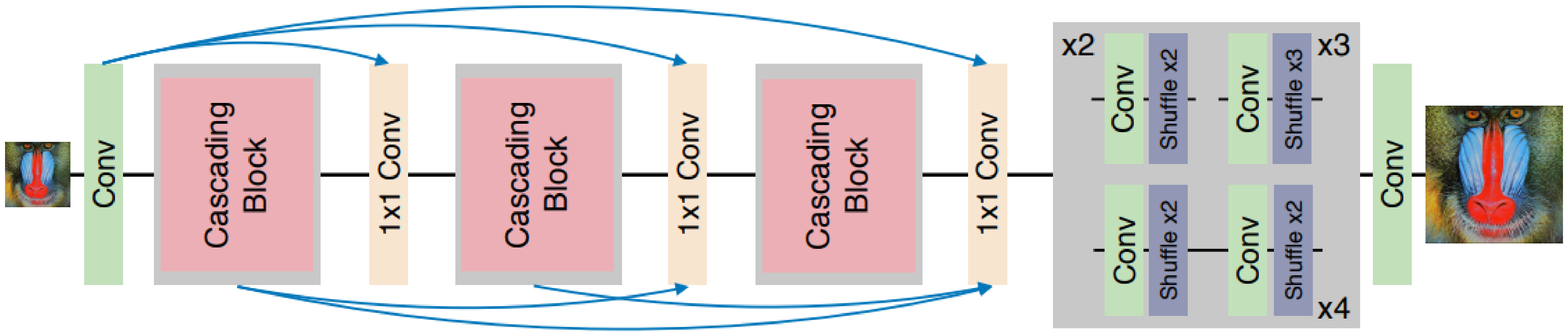


Accurate single image super-resolution using cascading dense connections

Deep CNNs based cascade structures for SISR

- CARN(Cascading Residual Network) [Efficiency]

1. Cascading Block improved the performance of SR
2. 1x1 Convolution reduce number of parameter
3. Efficient using Group convolution, can learn new feature

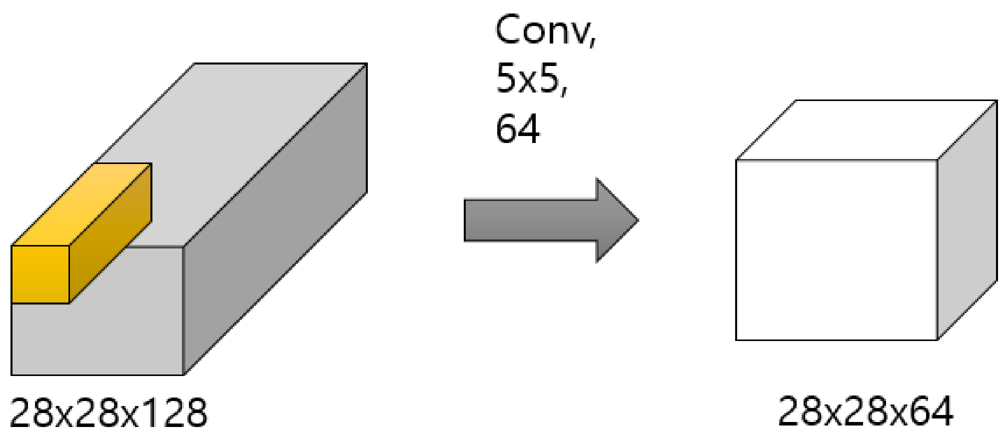


(b) Cascading Residual Network (CARN)

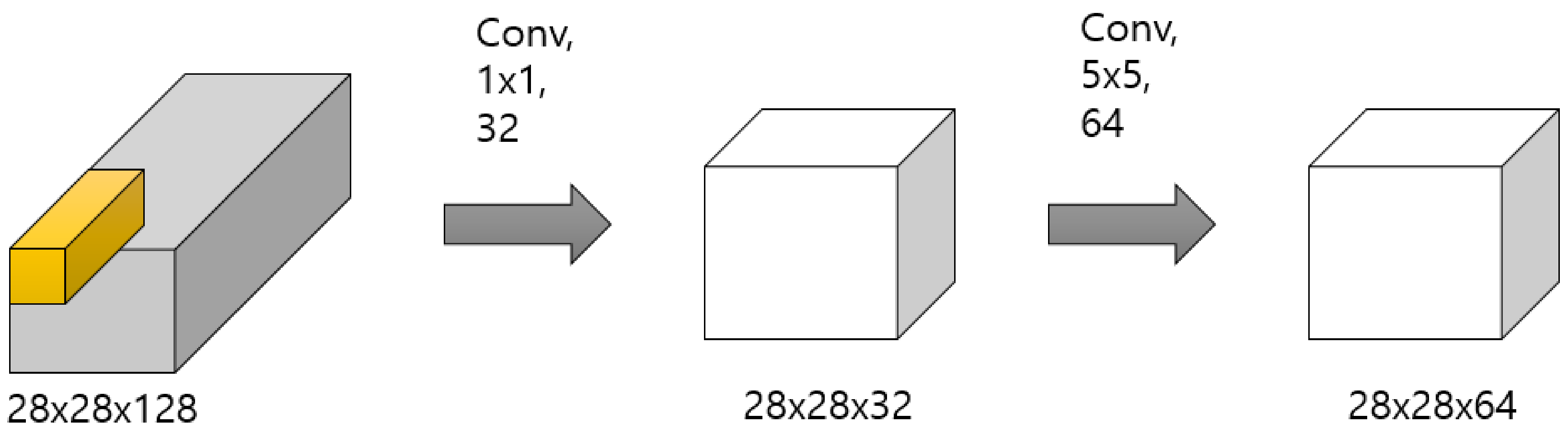
Fast, Accurate, and Lightweight Super-Resolution with cascading Residual Network

Deep CNNs based cascade structures for SISR

- 1x1 Convolution



#params = 28x28x64x5x5x128 = 160M



#params = 28x28x32x128x1x1 = 4.8M

#params = 28x28x64x5x5x32 = 40M

#total = 44.8M

Deep CNNs based blocks for SISR

Deep CNNs based Block used in computer vision tasks

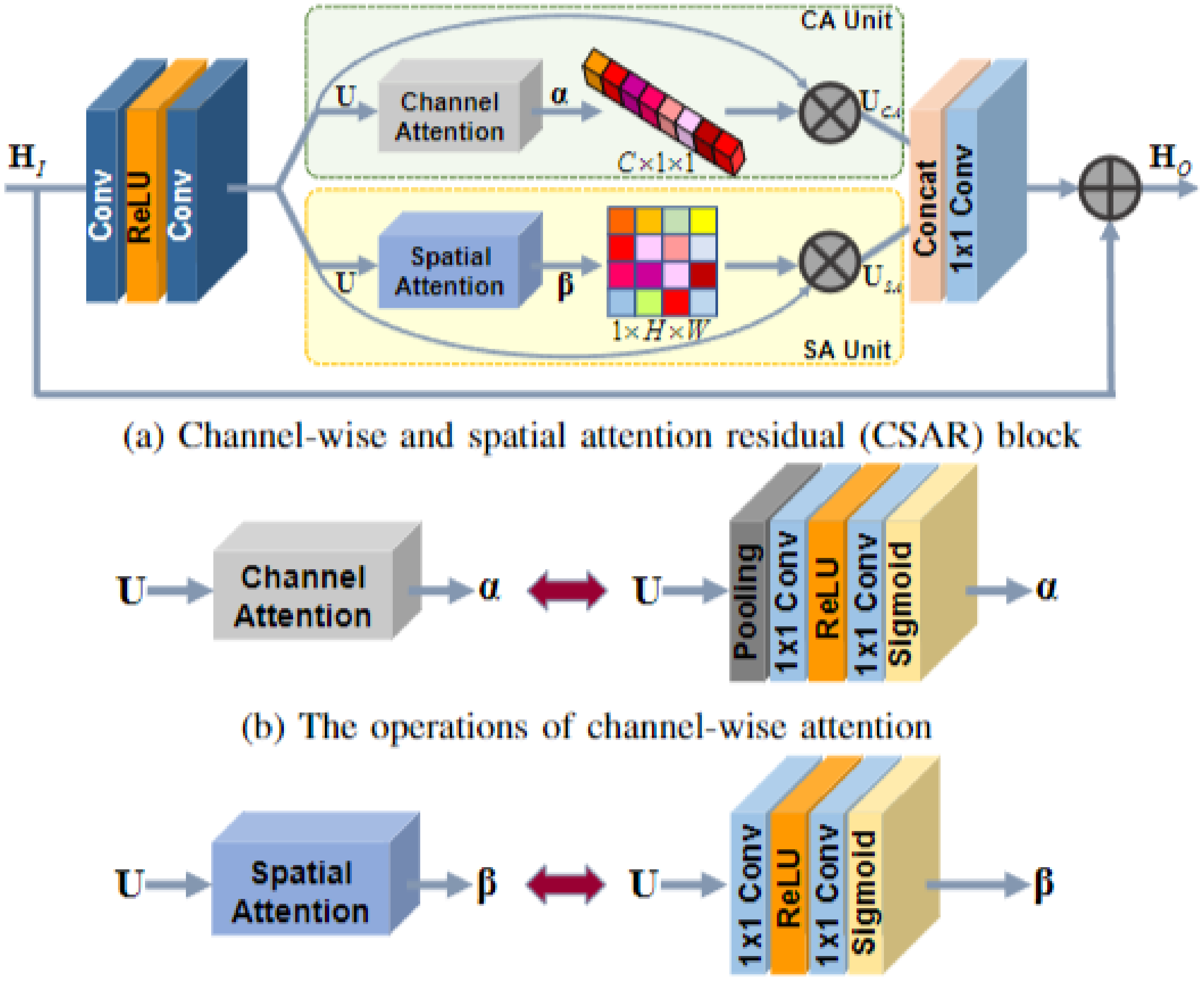
Better Performance
(Feature fusion method)

Lower Computational Cost
(Compressing network)

Deep CNNs based blocks for SISR

- CSAR(Channel-wise & Spatial Attention Residual) [Performance]

Combined Channel-wise and Spatial features



(a) Channel-wise and spatial attention residual (CSAR) block

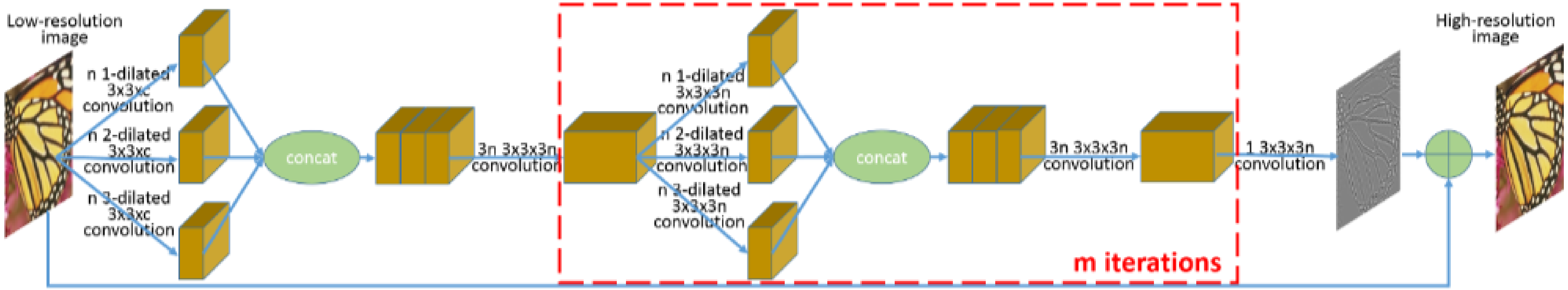
(b) The operations of channel-wise attention

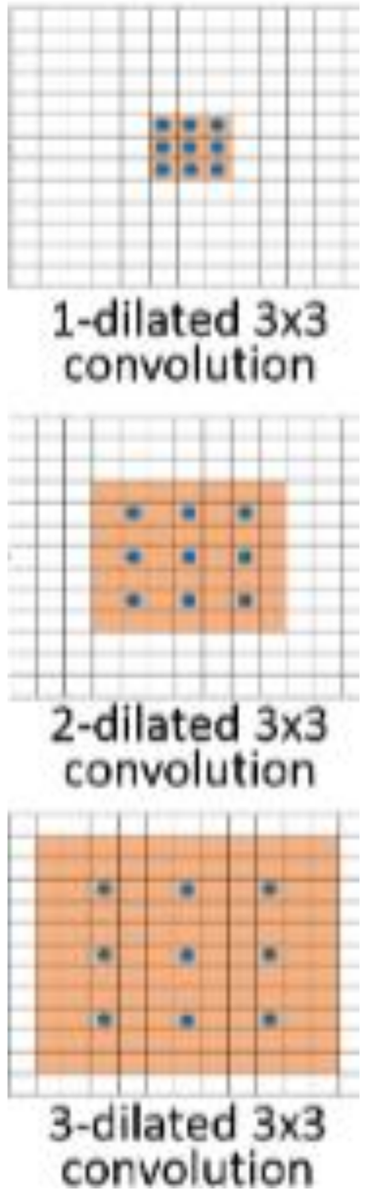
(c) The operations of spatial attention

Channel-wise and spatial feature modulation network for single image super-resolution

Deep CNNs based blocks for SISR

- Dilated Convolution [Performance]



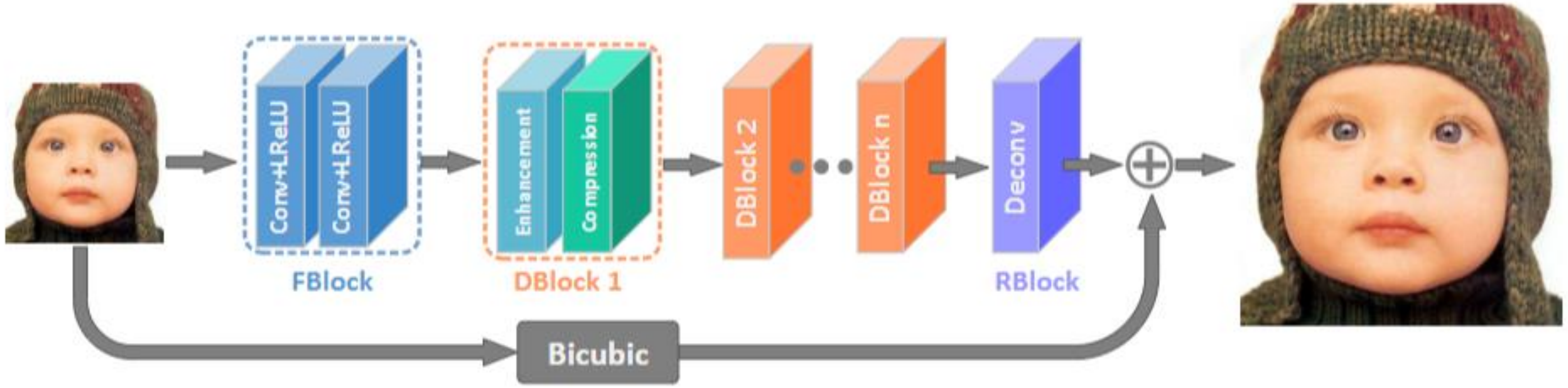


Dilated Convolution can get multi-scale information

Single image super-resolution with dilated convolution based multi-scale information learning inception module

Deep CNNs based blocks for SISR

- IDN(Information Distillation Network) [Low Computational Cost]
  - Group Convolution
  - 1x1 Convolution



김도완, 임채연

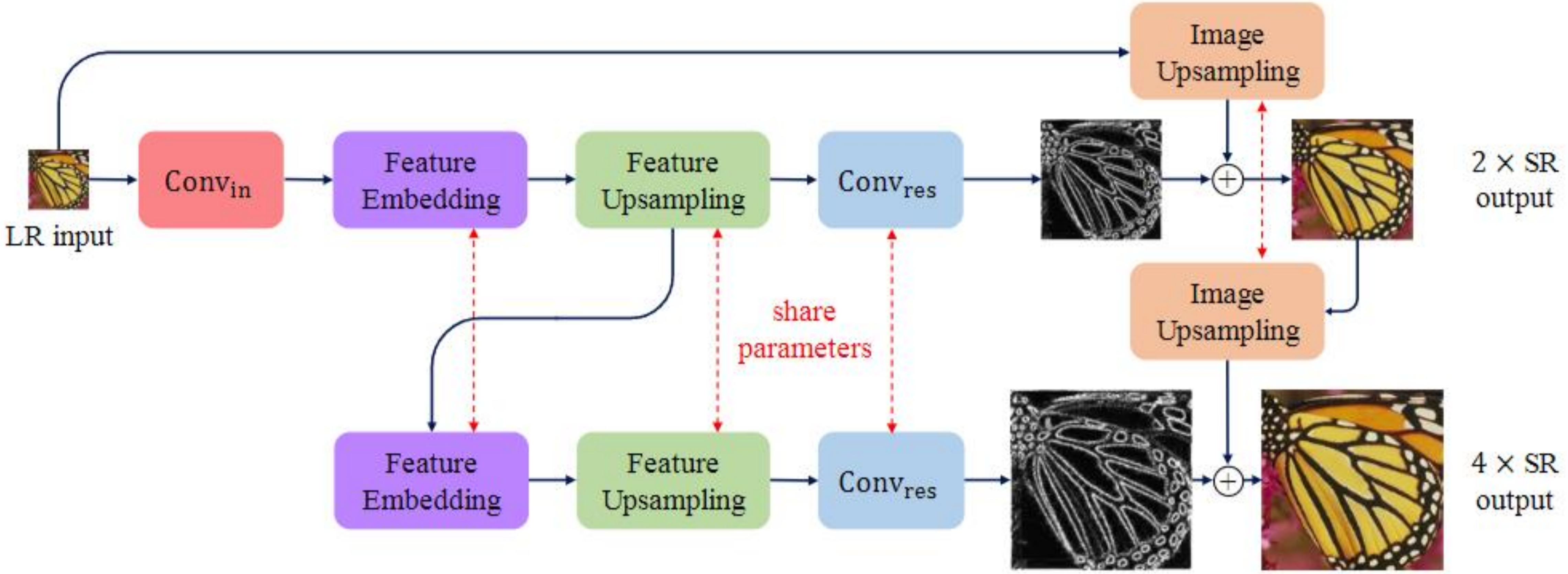
Fast and Accurate Single Image Super-Resolution via Information Distillation Network

Deep CNNs based blocks for SISR

- Laplacian Pyramid Network [Low Computational Cost]

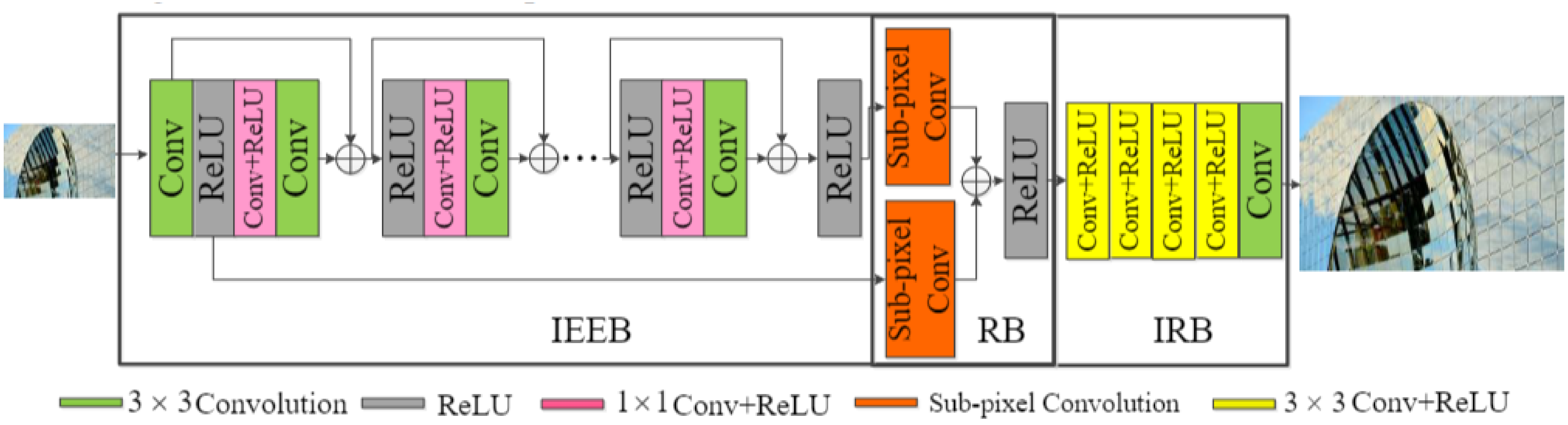
Use Parameter sharing

Decrease Parameter



Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks

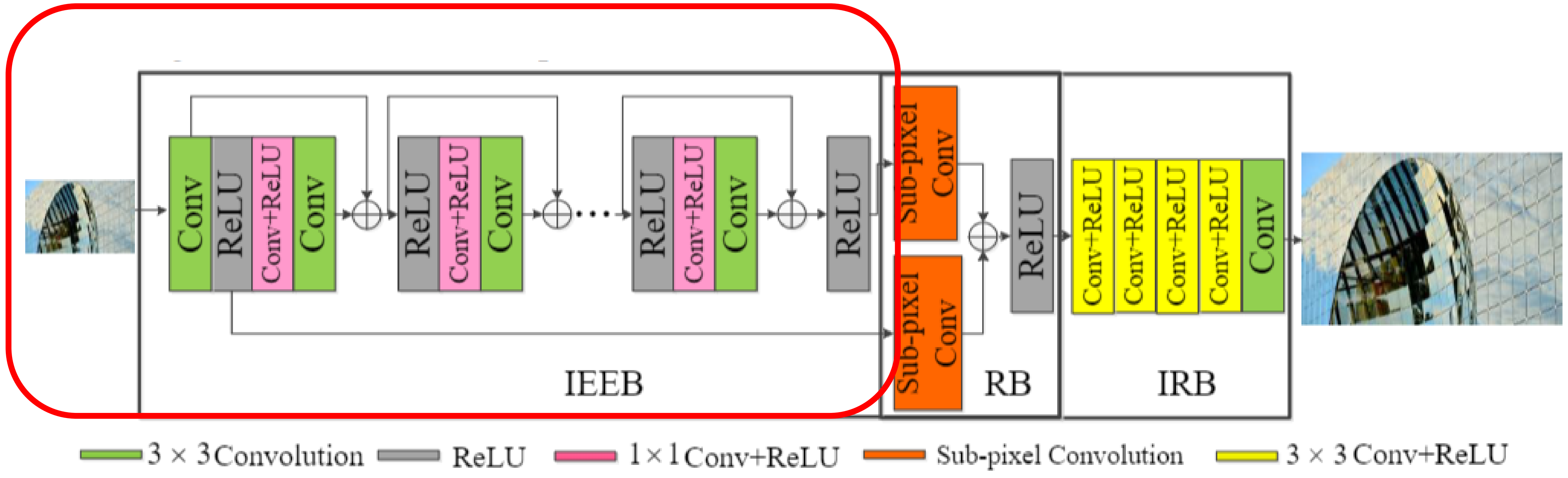
LESRCNN(Lightweight enhanced super-resolution CNN)



IEEB(Information Extraction and Enhancement Block)

Extract Low-frequency features

- Total 17 Convolution layers
- Two type of Convolutions
  - Odd layers: 3x3 Conv
  - Even layers: 1x1 Conv

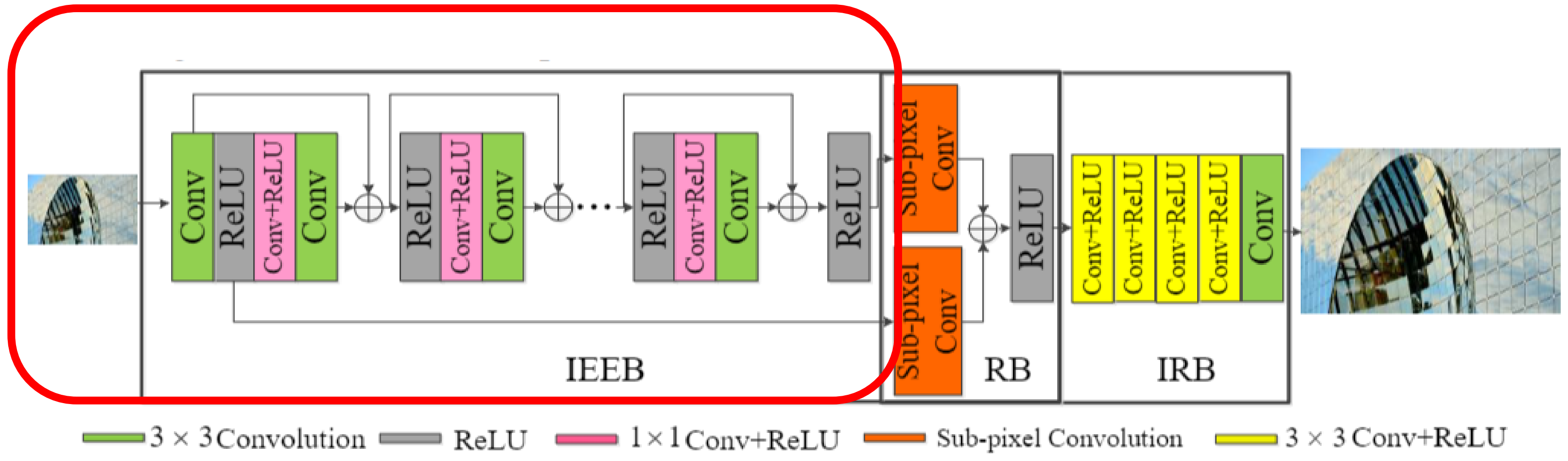


* O_i : output of i-th layer

$$O_c^i = \begin{cases} C_3(O_{i-1}) & i \text{ is odd} \\ C_1(O_{i-1}) & i \text{ is even} \end{cases} \Rightarrow O_j = \begin{cases} R(O_c^j + \sum_{j=1}^{j-2} O_c^j) & j \text{ is odd} \\ R(O_c^j) & j \text{ is even} \end{cases}$$

IEEB(Information Extraction and Enhancement Block)

```
x = self.sub_mean(x)          Mean shift
c0 = x
x1 = self.conv1(x)            3x3 Conv
x1_1 = self.ReLU(x1)
x2 = self.conv2(x1_1)         1x1 Conv+ReLU
x3 = self.conv3(x2)
x2_3 = x1+x3                  Residual
x2_4 = self.ReLU(x2_3)
x4 = self.conv4(x2_4)
x5 = self.conv5(x4)
x3_5 = x2_3 + x5
```



RB(Reconstruction Block)

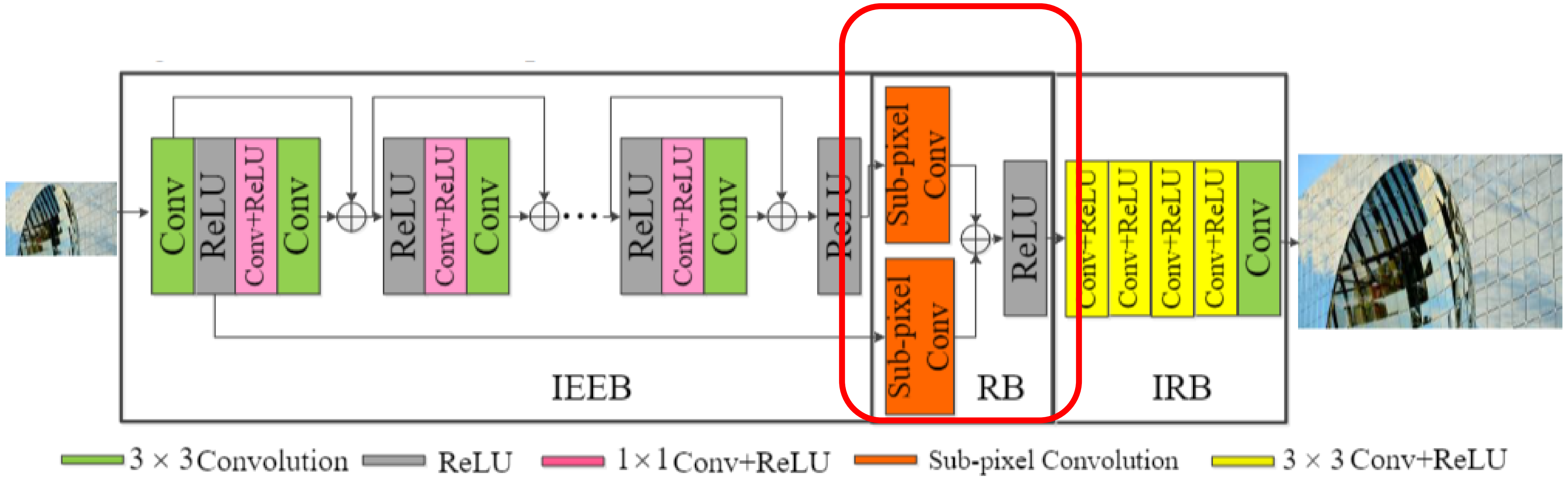
Upsampling Low-frequency Feature

- Upsample global & local features

  ➡ Low-frequency -> High-frequency

- Integrate sub-pixels output

  ➡ Enhance memory ability



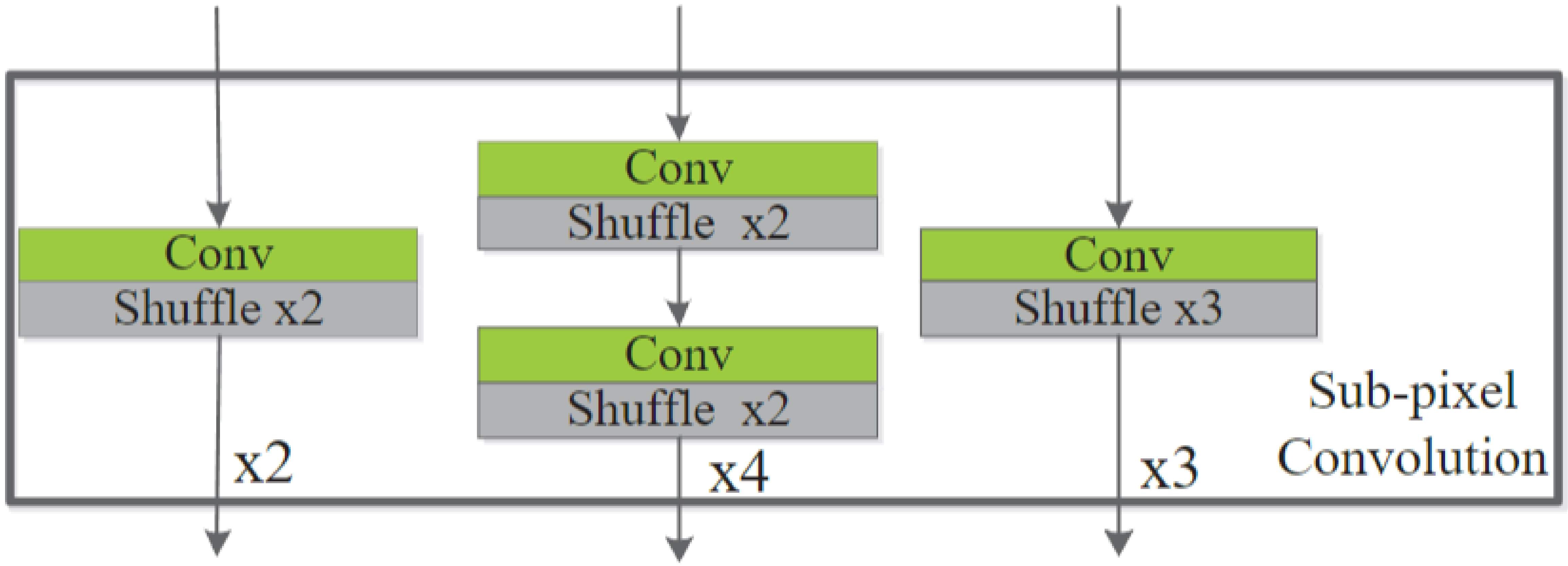
$*\ S(\cdot)$: sub-pixel Conv

$$O_{RB} = R(S(O_1) + S(O_{17}))$$

RB(Reconstruction Block)

- Sub-pixel Conv

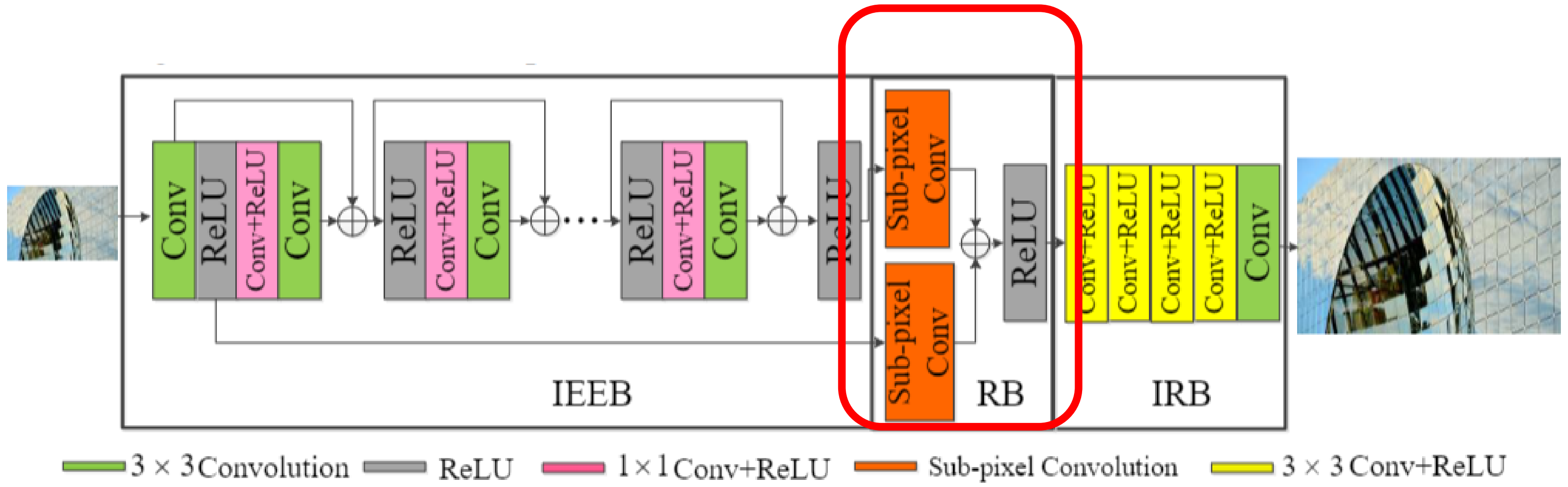
Divided into three types depending on scale



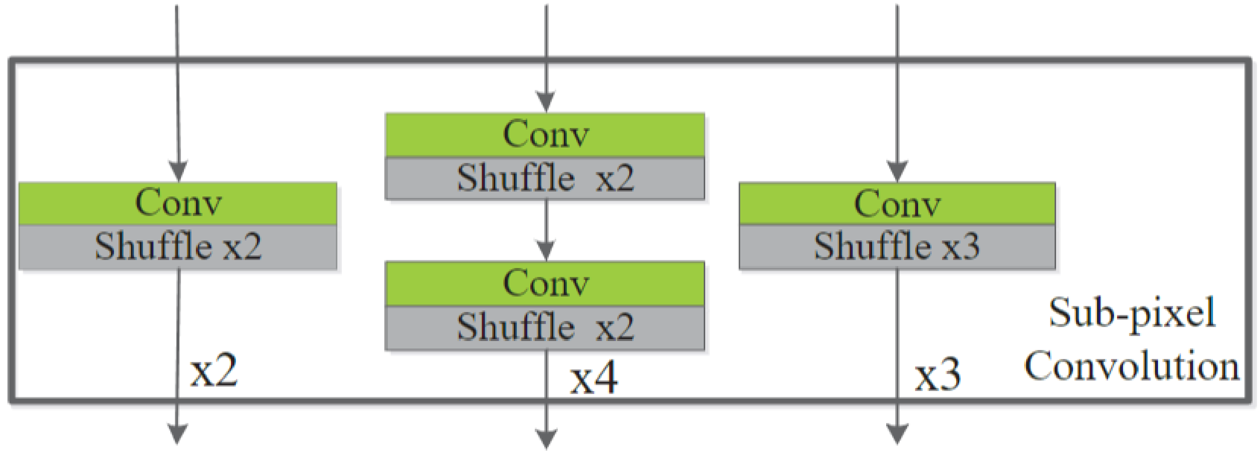
RB(Reconstruction Block)

```
temp = self.upsample(x17_3, scale=scale)
x1111 = self.upsample(x1_1, scale=scale) #tcw
temp1 = x1111+temp #tcw
temp2 = self.ReLU(temp1)
```

Sub-pixel Conv




```
if scale == 2 or scale == 4 or scale == 8:
    for _ in range(int(math.log(scale, 2))):
        #modules += [nn.Conv2d(n_channels, 4*n_channels, 3, 1, 1, groups=group), nn.ReLU(inplace=True)]
        modules += [nn.Conv2d(n_channels, 4*n_channels, 3, 1, 1, groups=group)]
        modules += [nn.PixelShuffle(2)]
elif scale == 3:
    #modules += [nn.Conv2d(n_channels, 9*n_channels, 3, 1, 1, groups=group), nn.ReLU(inplace=True)]
    modules += [nn.Conv2d(n_channels, 9*n_channels, 3, 1, 1, groups=group)]
    modules += [nn.PixelShuffle(3)]
```



IRB(Information Refinement Block)

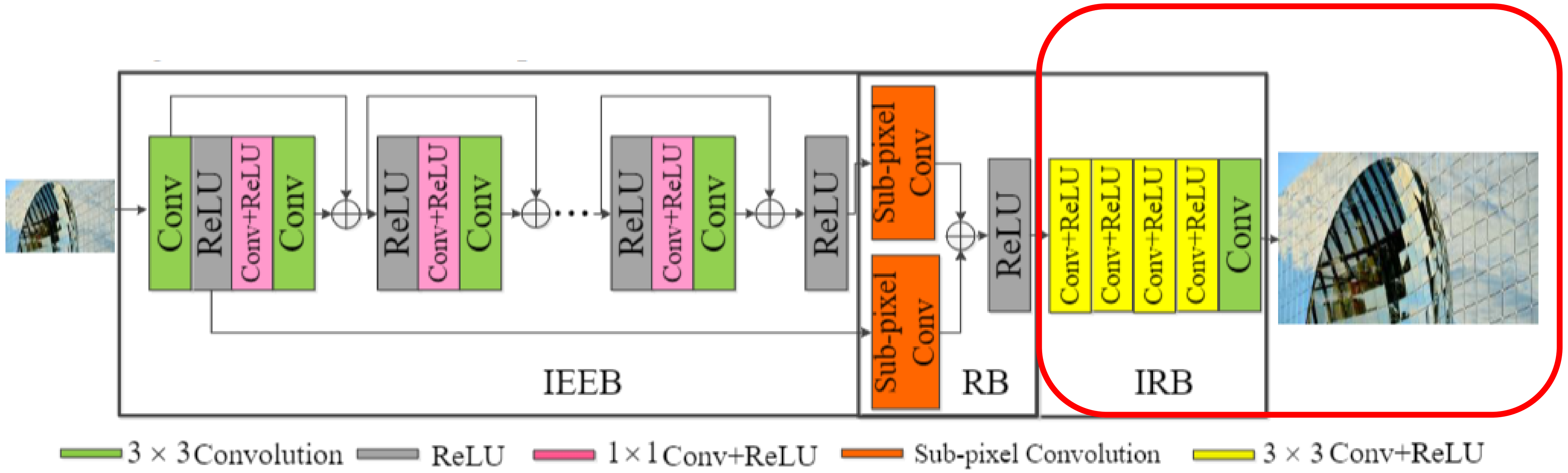
Learn more accuracy SR features

RB

High-frequency features

Conv + ReLU

SR features



3x3x64 Conv

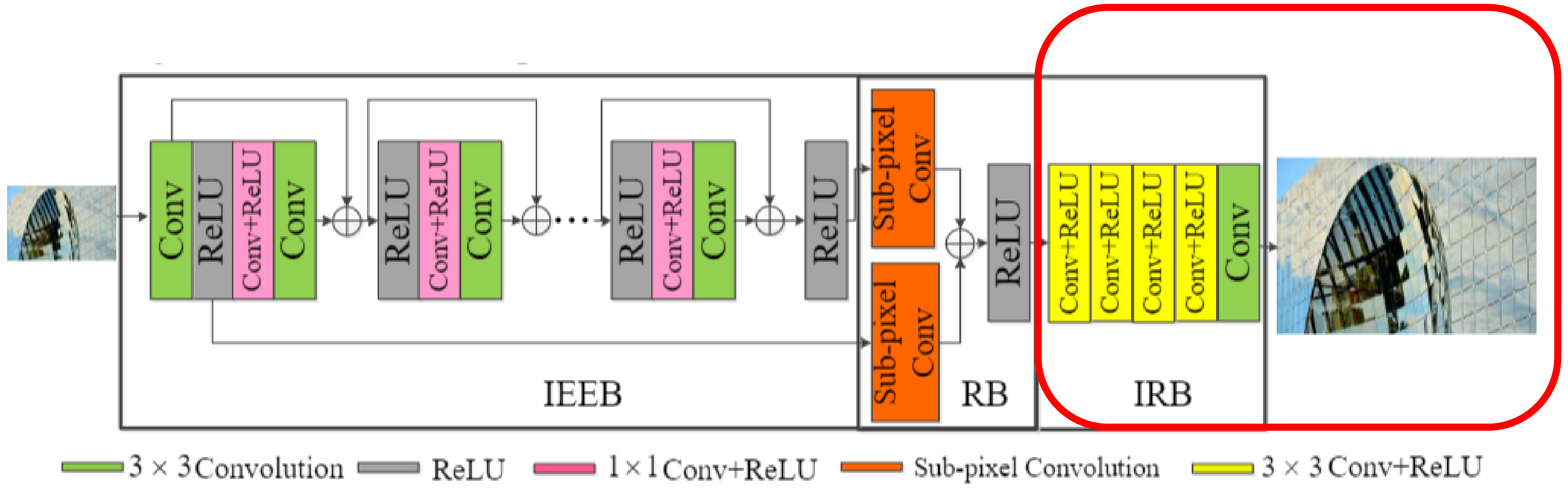
3x3x3 Conv

$$O_{SR} = C_3(R(C_3(R(C_3(R(C_3(R(C_3(O_{RB})))))))))$$

IRB(Information Refinement Block)

```
temp3 = self.conv17_1(temp2)
temp4 = self.conv17_2(temp3)
temp5 = self.conv17_3(temp4)
temp6 = self.conv17_4(temp5)
x18 = self.conv18(temp6)
out = self.add_mean(x18)
```

Output channel: 3



Loss function

MSE(Mean Squared Error)

I_{LR}^{i} : i-th low resolution image

I_{HR}^{i} : i-th high resolution image

T : total number of training image

$$l(p) = \frac{1}{2T}\sum_{i=1}^{T}\left\|f_{LESRCNN}(I_{LR}^{i}) - I_{HR}^{i}\right\|^{2}$$

DataSet

Type	Name	Explanation
Train	DIV2K	800 training, 100 validation, 100 test color images. (x2, x3, x4) Cropped 64 x 64
Test	Set5	5 color images (x2, x3, x4)
	Set14	14 color Images (x2, x3, x4)
	BSD100	100 color images (x2, x3, x4)
	Urban100	100 color images (x2, x3, x4)

Convert to
Y channel(YCbCr)

Evaluation Formula

PSNR(Peak Signal-to-Noise Ratio)

R : maximum value of pixel

$$PSNR = 10 \log_{10}\left(\frac{R^2}{MSE}\right)$$

→ Independent of human visual quality

Evaluation Formula

SSIM(Structural Similarity Index Map)

l : Luminance
C : Contrast
S : Structural

$$SSIM(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma}$$

Expressing human visual quality

Training

Hyper Parameter	Value		
Batch size	64		
Epoch	6e+5(600000)		
Optimizer	Adam	Beta1	0.9
		Beta2	0.999
		epsilon	1e-8(0.0000001)
LR Scheduler	Initial 1e-4(0.0001) -> Halved every 4e+5(400000) steps		

Ablation Study

Scale	Methods	Set5
		PSNR/SSIM
$\times 4$	SN	31.64/0.8864
	HN	31.62/0.8852
	IEEB	31.73/0.8877
	IEEB+RB	31.76/0.8881
	LESRCNN	31.88/0.8903

(1). Average PSNR and SSIM of different methods

Sizes	Methods	
	SN	HN
	$\times 4$	
256×256	0.00669	0.00651
512×512	0.00879	0.00869
1024×1024	0.01672	0.01651

(2). Running time of two methods at different image size

Methods	Parameters	Flops
SN	630K	3.06G
HN	368K	1.38G

(3). Complexity of two comparative methods

Ablation Study

Dataset	Model	×2	×3	×4
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
U100	Bicubic	26.88/0.8403	24.46/0.7349	23.14/0.6577
	A+ [54]	29.20/0.8938	26.03/0.7973	24.32/0.7183
	JOR [10]	29.25/0.8951	25.97/0.7972	24.29/0.7181
	RFL [41]	29.11/0.8904	25.86/0.7900	24.19/0.7096
	SelfEx [19]	29.54/0.8967	26.44/0.8088	24.79/0.7374
	DnCNN [69]	30.74/0.9139	27.15/0.8276	25.20/0.7521
	TNRD [8]	29.70/0.8994	26.42/0.8076	24.61/0.7291
	FDSR [33]	30.91/0.9088	27.23/0.8190	25.27/0.7417
	SRCNN [11]	29.50/0.8946	26.24/0.7989	24.52/0.7221
	FSRCNN [12]	29.88/0.9020	26.43/0.8080	24.62/0.7280
	VDSR [22]	30.76/0.9140	27.14/0.8279	25.18/0.7524
	DRCN [23]	30.75/0.9133	27.15/0.8276	25.14/0.7510
	LapSRN [26]	30.41/0.9100	-	25.21/0.7560
	MemNet [47]	31.31/0.9195	27.56/0.8376	25.50/0.7630
	CARN-M [2]	31.23/0.9193	27.55/0.8385	25.62/0.7694
	WaveResNet [5]	30.96/0.9169	27.28/0.8334	25.36/0.7614
	CPCA [59]	28.17/0.8990	25.61/0.8123	23.62/0.7257
	NDRCN [7]	31.06/0.9175	27.23/0.8312	25.16/0.7546
	LESRCNN (Ours)	31.45/0.9206	27.70/0.8415	25.77/0.7732
	LESRCNN-S (Ours)	31.45/0.9207	27.76/0.8424	25.78/0.7739

(PSNR and SSIM of different techniques on U100)

Ablation Study

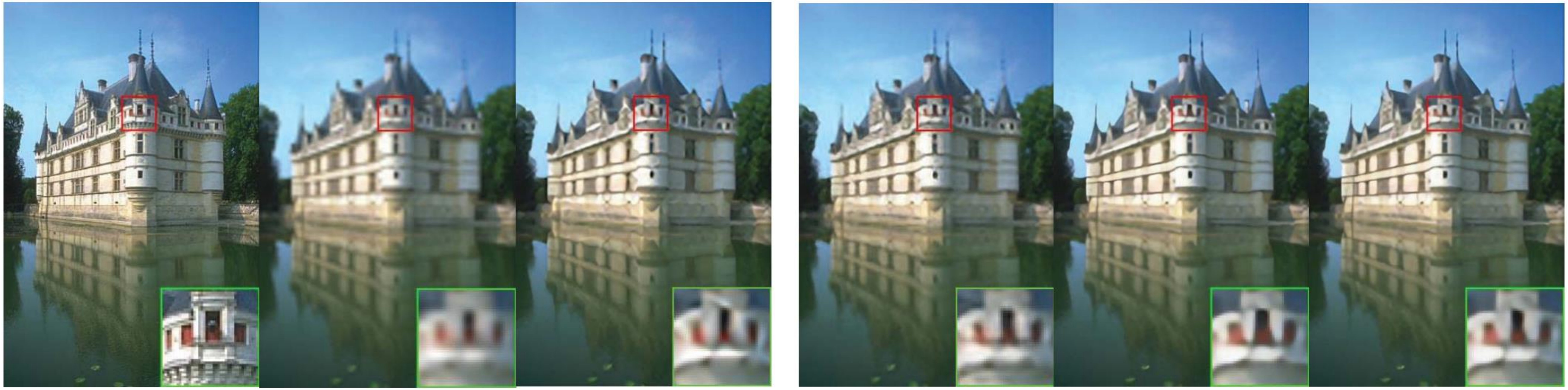
Single Image Super-Resolution				
Size	VDSR [22]	MemNet [47]	CARN-M [2]	LESRCNN (Ours)
256×256	0.0172	0.8774	0.0159	0.0102
512×512	0.0575	3.605	0.0199	0.0129
1024×1024	0.2126	14.69	0.0320	0.0222

(1). Running time of four networks at different image size

Methods	Parameters	Flops
VDSR [22]	665K	10.90G
DnCNN [69]	556K	9.18G
DRCN [23]	1774K	29.07G
MemNet [47]	677K	11.09G
LESRCNN (Ours)	516K	3.08G

(2). Complexity of five networks

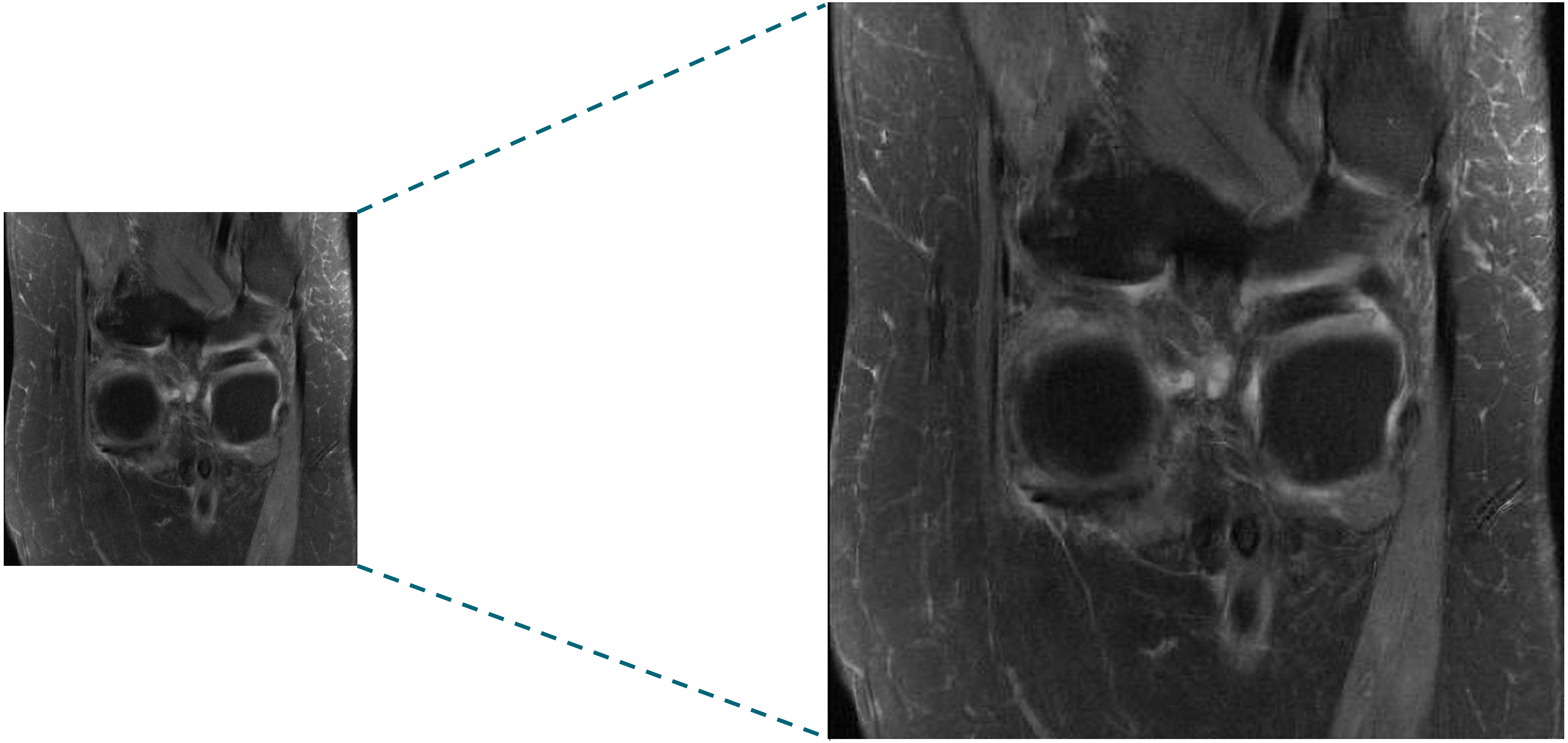
Ablation Study



(1). HR image (PSNR/SSIM) (2). Bicubic (25.26/0.7539) (3). SelfEx (25.83/0.7852) (4). SRCNN (25.78/0.7767) (5). CARN-M (26.39/0.8046) (6). LESRCMM (26.46/0.8061)

Visual effects of different methods (X4 scale)

Ablation Study



(LESRCNN Example)

Conclusion

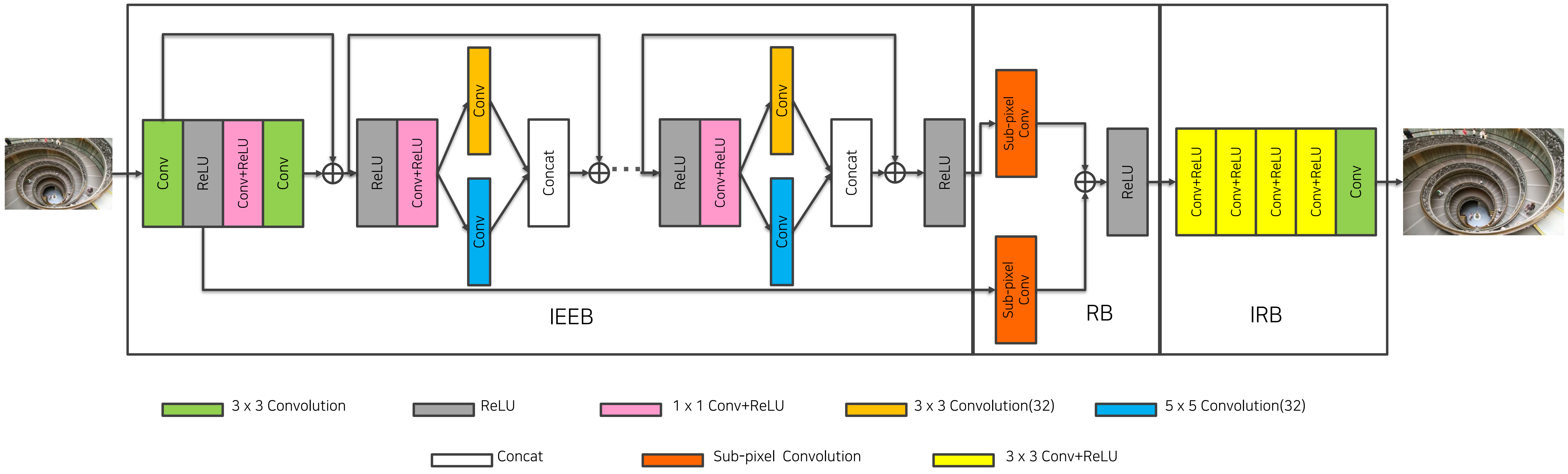
- IEEB: Extract low-frequency features, reduce the number of parameters
- RB: Convert low-frequency features into high-frequency features
- IRB: high-frequency features -> more accurate SR features

➡ **Low parameters & High performance**

Current Situation

Train hyper parameter	Urban100 (PSNR/SSIM)	
	Original	Upgrade
Epochs: 200000, Decay: 150000	31.14/0.9171	31.30/0.9191
Epochs: 600000, Decay: 400000	31.45/0.9206	31.61/0.9226

Current Situation



31.45/0.9206 → 31.61/0.9226

516K → 707K

Result

모델	Urban 100		
	x2	x3	X4
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
LESRCNN	31.45/0.9206	27.70/0.8415	25.77/07732
제안한 LESRCNN	31.61/0.9226	27.84/0.8442	25.81/0.7756

감사합니다!

질문이 있으신가요?