Local Texture Estimator for Implicit Representation Function

Jaewon Lee Kyong Hwan Jin
Image Processing Laboratory (IPL), DGIST, Korea

Paper Code

Introduction

Single image super-resolution (SR)

We need to train and store several models for each scale factor, when an upsampler is implemented by sub-pixel convolution

Arbitrary-scale SR

Arbitrary-scale SR methods pave the way to restore images in a continuous manner with **single network**

LR X1.3 X1.7 X2.2 X2.8

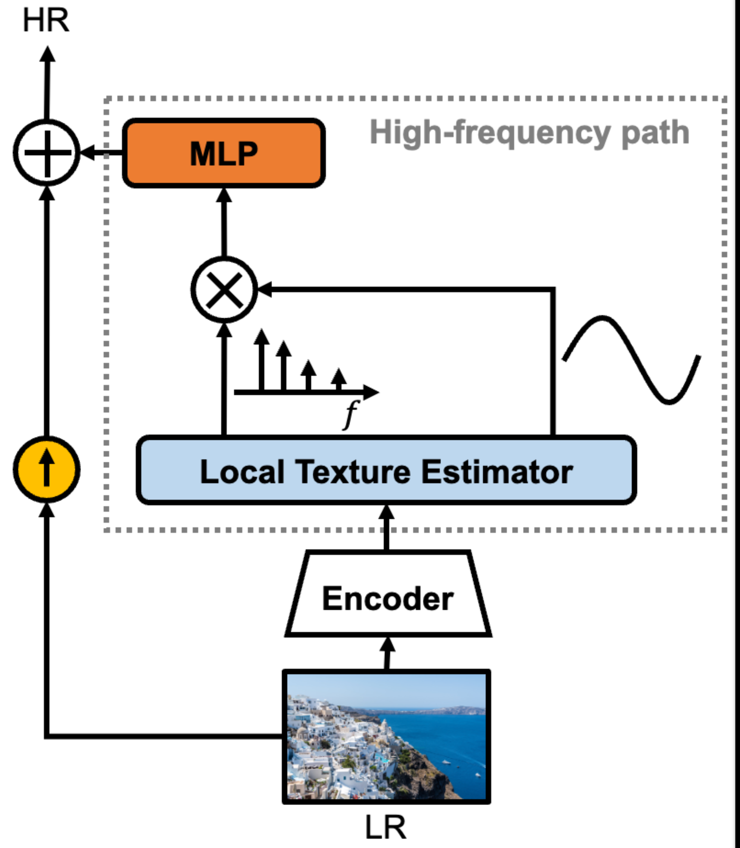
LR Bicubic (X32) Ours (X32)

Implicit representation function

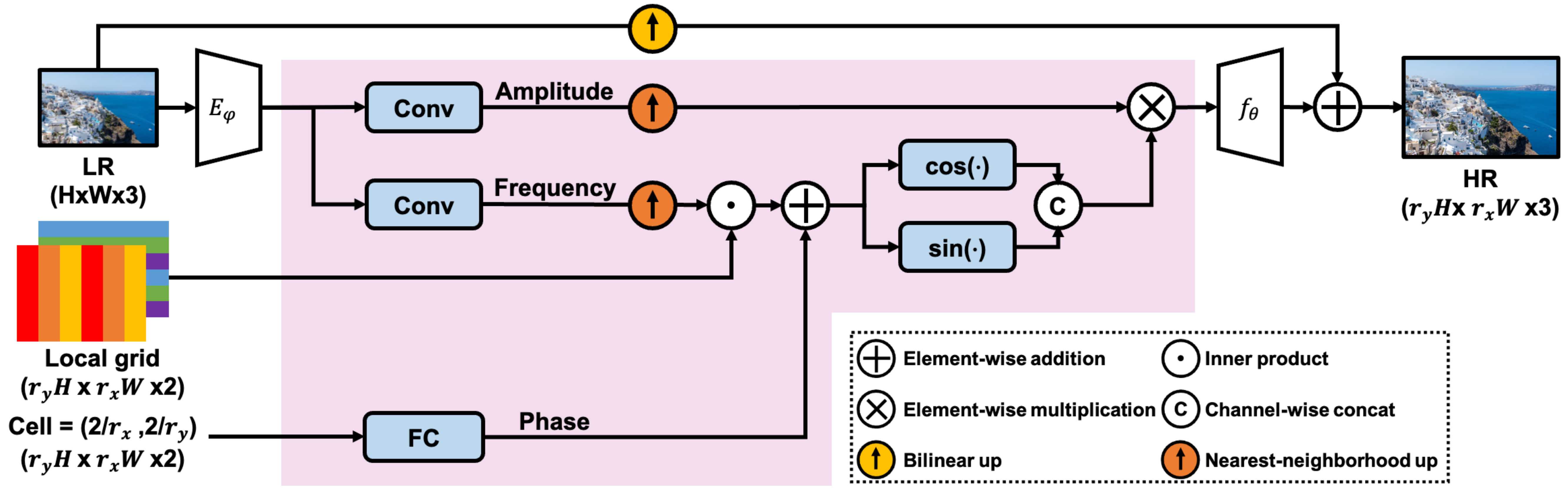
Implicit representation function shed light on representing images in arbitrary resolution

However, a standalone MLP shows limited performance in learning **high-frequency Fourier coefficients**

Hence, we study arbitrary-scale SR through the lens of **Fourier analysis**



Method



Local implicit representation function

$$\mathbf{I}^{\mathbf{HR}}[\mathbf{x};\boldsymbol{\theta}] = \sum_{j\in\mathcal{T}} w_j f_\theta(\mathbf{z}_j, \mathbf{x}-\mathbf{x}_j, \mathbf{c}) \quad \text{where } \mathbf{z} = E_\varphi(\mathbf{I}^{\mathbf{LR}})$$

- Implicit neural function represents image **continuously**
- However, an MLP suffers from **spectral bias problem**

Learning dominant frequency component

Local Texture Estimator

$$\mathbf{I}^{\mathbf{HR}}[\mathbf{x};\boldsymbol{\theta},\boldsymbol{\psi}] = \sum_{j\in\mathcal{T}} w_j f_\theta\left(h_\psi(\mathbf{z}_j, \mathbf{x}-\mathbf{x}_j, \mathbf{c})\right) + \mathbf{I}_{\uparrow}^{\mathbf{LR}}[\mathbf{x}]$$

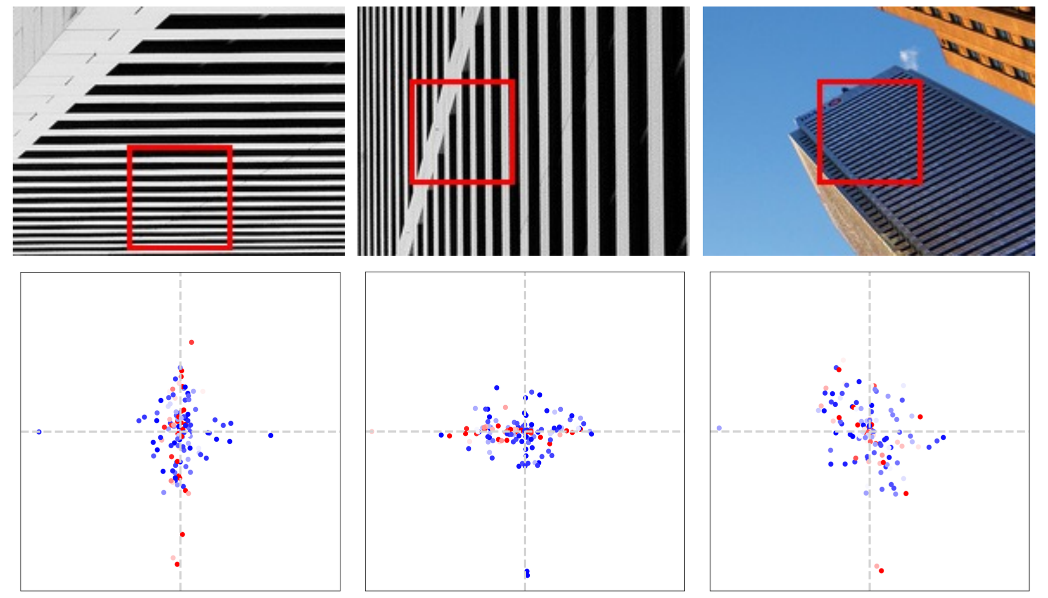
where h_ψ is a local texture estimator

Amplitude & Frequency & Phase

$h_a(\cdot): \mathbb{R}^C \mapsto \mathbb{R}^{2K}$ (Amplitude) $h_f(\cdot): \mathbb{R}^C \mapsto \mathbb{R}^{K\times 2}$ (Frequency) $h_p(\cdot): \mathbb{R}^2 \mapsto \mathbb{R}^K$ (Phase)

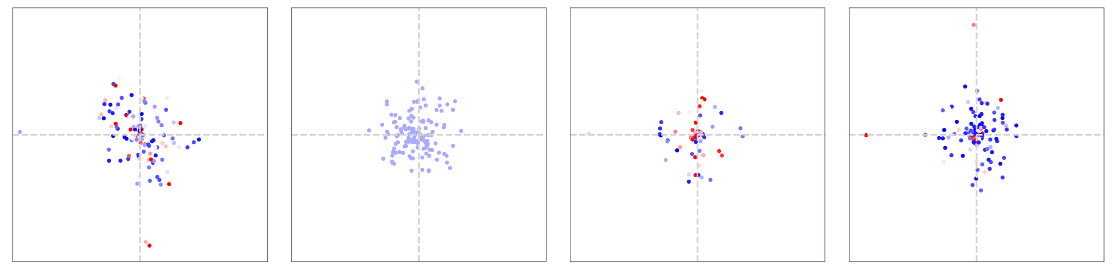
$$h_\psi(\mathbf{z}_j, \boldsymbol{\delta}, \mathbf{c}) = \mathbf{A}_j \odot \begin{pmatrix} \cos\left(\pi\left(\mathbf{F}_j\boldsymbol{\delta} + h_p(\hat{c})\right)\right) \\ \sin\left(\pi\left(\mathbf{F}_j\boldsymbol{\delta} + h_p(\hat{c})\right)\right) \end{pmatrix}$$

where $\mathbf{A}_j = h_a(\mathbf{z}_j)$ $\mathbf{F}_j = h_f(\mathbf{z}_j)$ $\hat{c} = \max(c, c_{tr})$



Horizontal Vertical Diagonal

LTE characterizes image textures in 2D Fourier space



LTE LTE (-A) LTE (-F) LTE (-P)

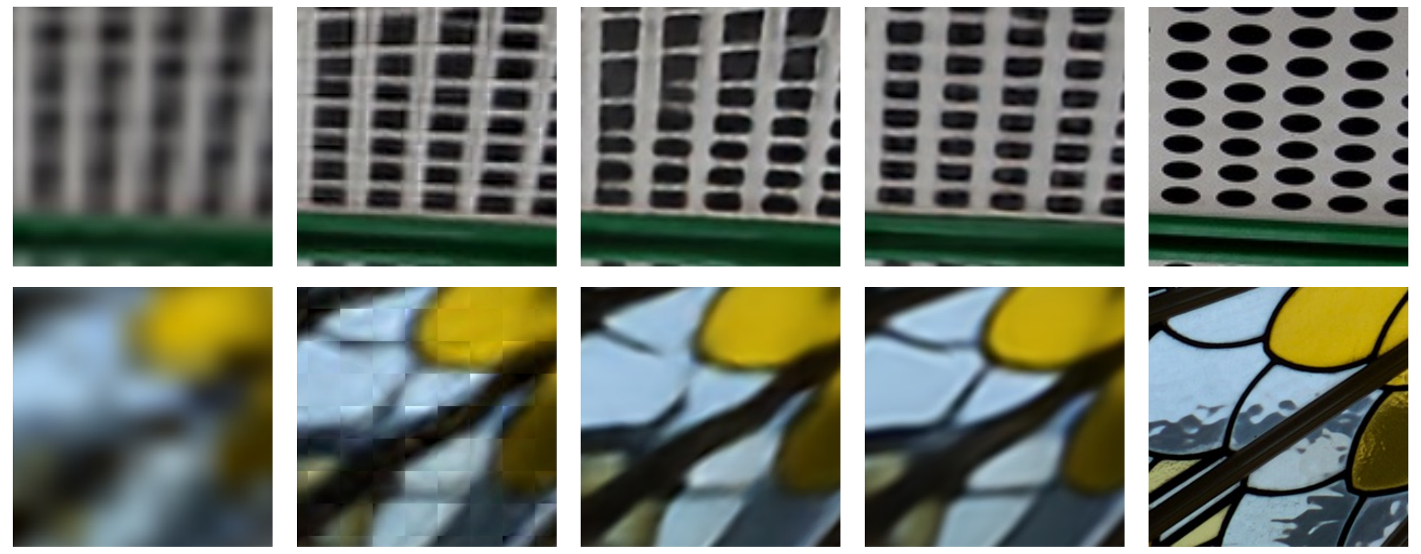
Each component is significant in estimating Fourier space

Results

Quantitative comparison

Method	In-scale ×2	×3	×4	Out-of-scale ×6	×12	×18	×24	×30
Bicubic [15]	31.01	28.22	26.66	24.82	22.27	21.00	20.19	19.59
EDSR-baseline [15]	34.55	30.90	28.94	-	-	-	-	-
EDSR-baseline-MetaSR [4, 9]	34.64	30.93	28.92	26.61	23.55	22.03	21.06	20.37
EDSR-baseline-LIIF [4]	34.67	30.96	29.00	26.75	23.71	22.17	21.18	20.48
EDSR-baseline-LTE (ours)	34.72	31.02	29.04	26.81	23.78	22.23	21.24	20.53
RDN-MetaSR [4, 9]	35.00	31.27	29.25	26.88	23.73	22.18	21.17	20.47
RDN-LIIF [4]	34.99	31.26	29.27	26.99	23.89	22.34	21.31	20.59
RDN-LTE (ours)	35.04	31.32	29.33	27.04	23.95	22.40	21.36	20.64
SwinIR-MetaSR† [4, 9]	35.15	31.40	29.33	26.94	23.80	22.26	21.26	20.54
SwinIR-LIIF† [4]	35.17	31.46	29.46	27.15	24.02	22.43	21.40	20.67
Swinir-LTE (ours)	35.24	31.50	29.51	27.20	24.09	22.50	21.47	20.73

Qualitative comparison



Bicubic MetaSR LIIF LTE GT

Blurry Blocky artifacts Structural distortion High-frequency details

Model complexity

#Eval/Query	Method	# Params.	Mem. (GB)	Time (ms)
9216 (96 × 96)	MetaSR [9]	1.7M	1.9	3462
	LIIF [4]	1.6M	1.9	4559
	LTE (ours)	1.7M	2.3	2912
1.6M (1248 × 1248)	MetaSR [9]	1.7M	*OOM*	-
	LIIF [4]	1.6M	11.4	873
	LTE (ours)	1.7M	10.2	925
	LTE+ (ours)	1.7M	7.1	483

Conclusion

LTE-based neural function : Fourier information + MLP
➔ Arbitrary-scale SR with high-frequency details