Meta-Transfer Learning for Zero-Shot Super-Resolution

Presented in CVPR 2020
Authors - Jae Woong Soh, Sunwoo Cho, Nam Ik Cho (Seoul National University)

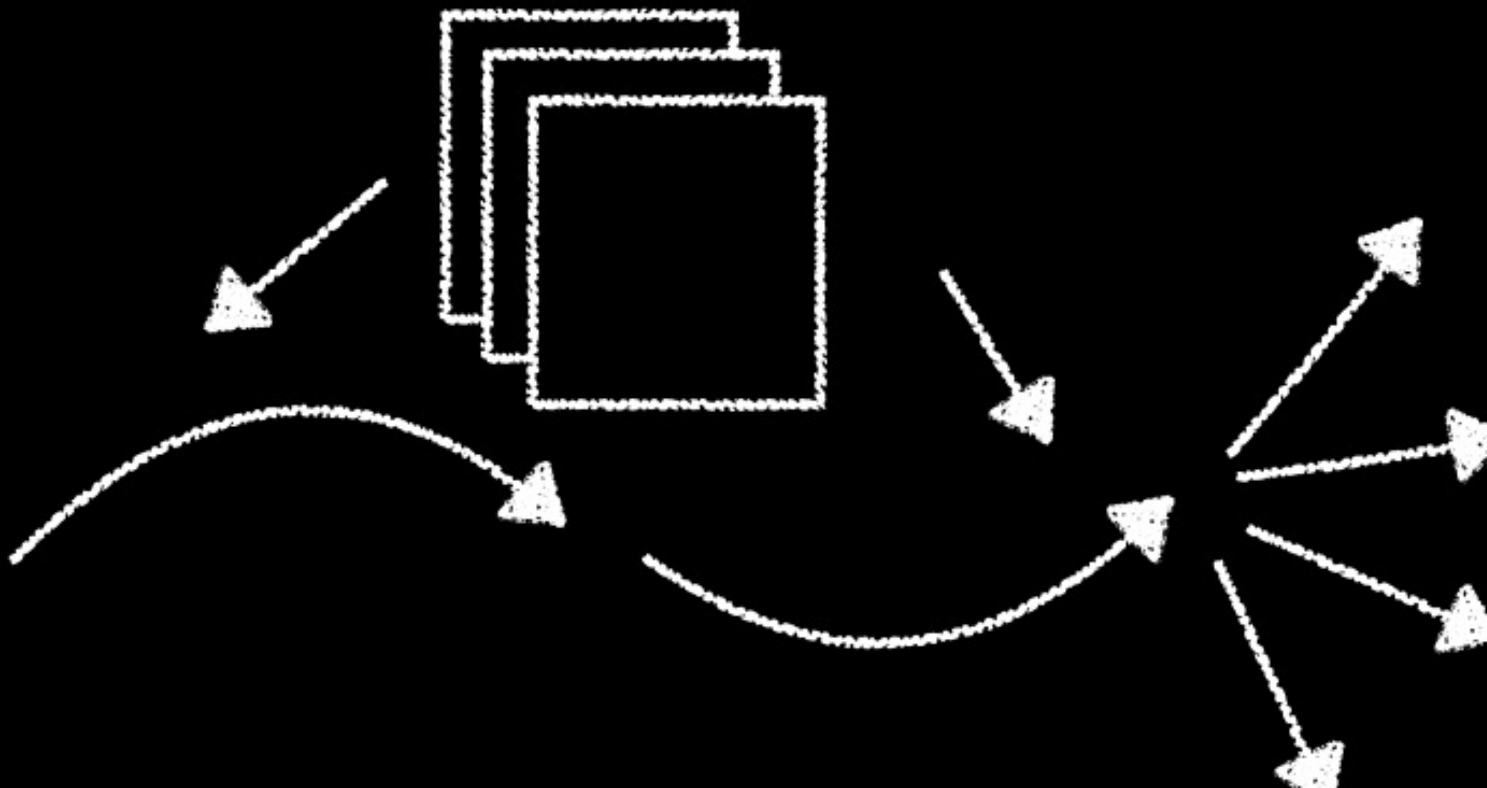


Image Processing Team
Seonok Kim

Introduction

Previous Methods

Proposed Methods

Experiments

Discussion

Conclusion

Introduction

MZSR (Meta Transfer Learning for Zero Shot Super Resolution)

Recent CNN based methods

Trained on large-scale ‘external’ dataset

Applicable only to the specific condition of data that they are supervised
(e.g., “bicubic” downsampled noise-free image)

→

Zero Shot Super Resolution

Proposed for flexible internal learning.

Transfer-Learning

Employed a pre-trained network with meta-learning method

Meta-Learning

MAML[1] scheme for fast adaptation of ZSSR

→

Few gradient updates

One single gradient update can yield quite considerable results.

Flexibility

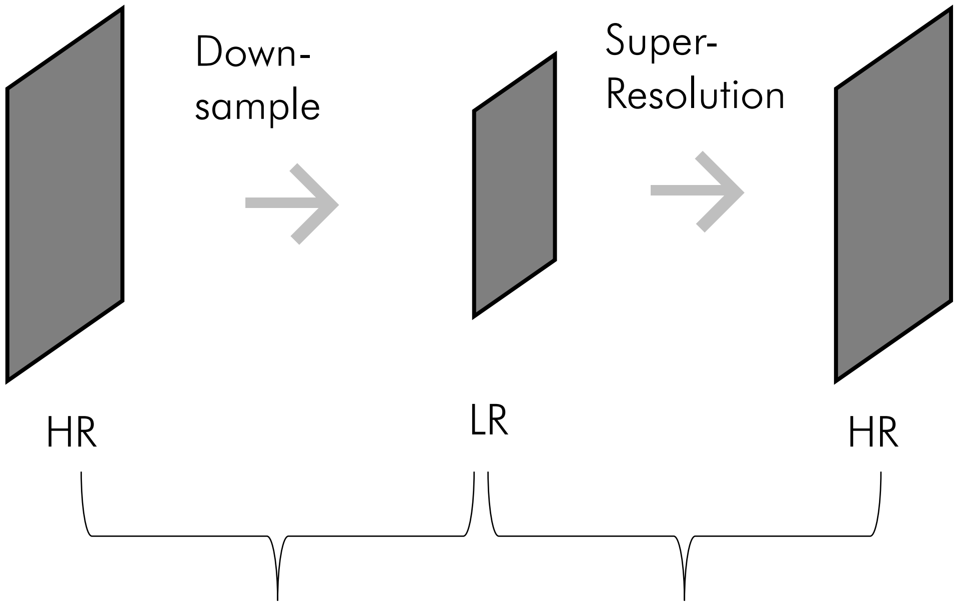
It can be applied to real-world scenes.

1) Model-Agnostic Meta-Learning (https://arxiv.org/abs/1703.03400)

Previous Methods

Background

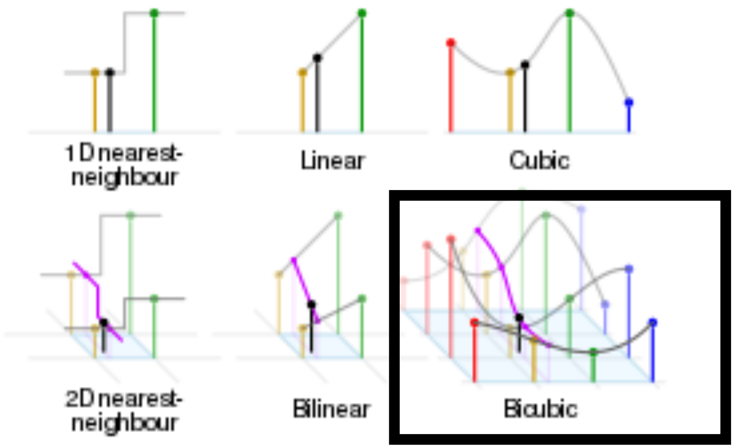
Why Down-sample images



LR- HR pairs is used
in the training phase

The trained network
used for SR
in the test phase

Bicubic Down-sampling [2)]

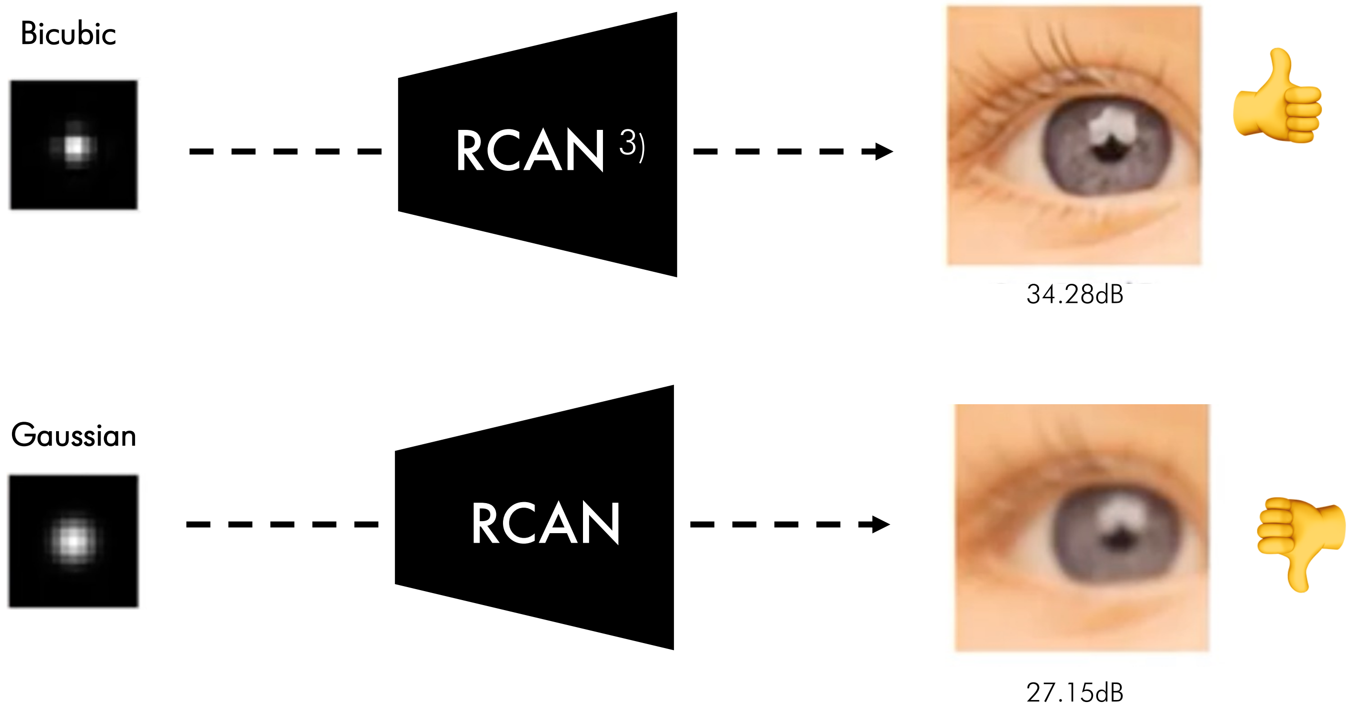


A neighborhood of 4X4 pixel is
considered to perform the interpolation

2) Comparison of bicubic interpolation with some 1- and 2-dimensional interpolations (Wikipedia).

CNN-based Methods

Convolutional neural networks (CNNs) have shown dramatic improvements in single image super-resolution (SISR) by using large-scale external samples.



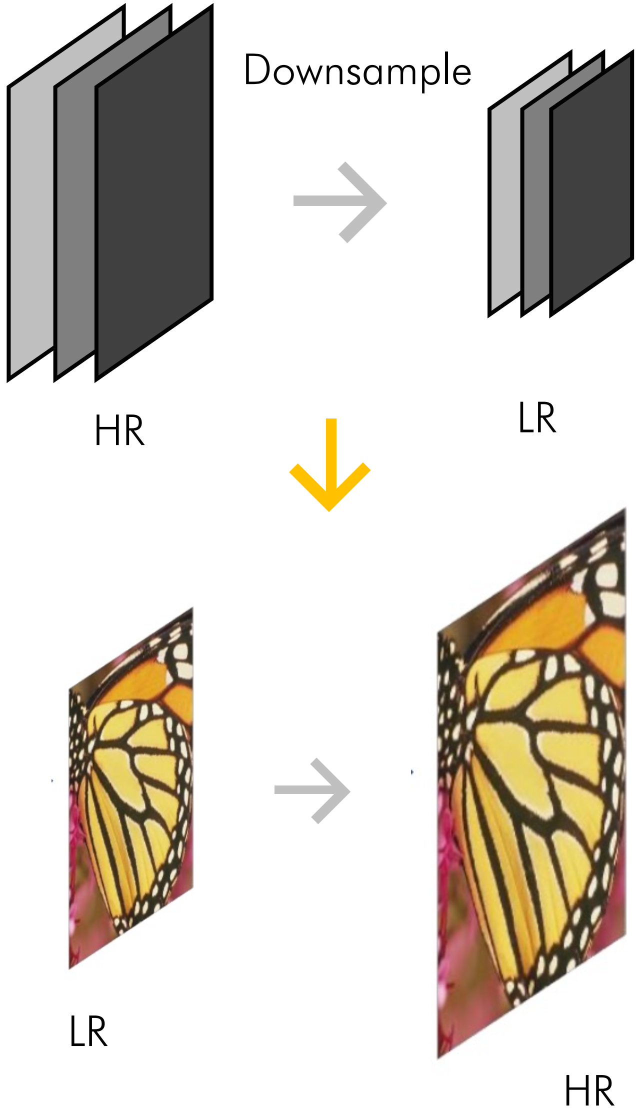
Limitations

Most of the recent CNN-based methods are applicable only to the <u>specific condition</u> of data that they are supervised.
(e.g., "bicubic" downsampling.)

They are rained on large-scale external dataset and the number of parameters are large to boost performance

3) ECCV '18 Image Super-Resolution Using Very Deep Residual Channel Attention Networks (https://arxiv.org/abs/1807.02758)

SISR (Single Image Super-Resolution)



SISR is to find a plausible high-resolution image from its counterpart low-resolution image.

Low Resolution Image

Blur kernel

Gaussian noise

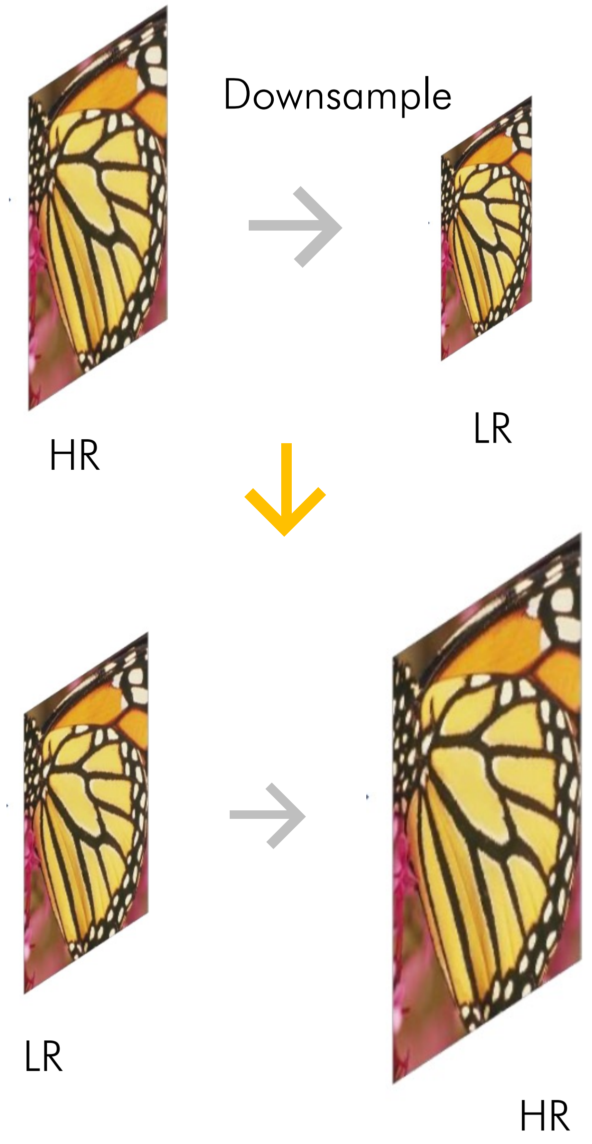
$$\mathbf{I}^{\mathbf{k}}_{LR} = (\mathbf{I}_{HR} * \mathbf{k}) \downarrow_s + \mathbf{n}$$

Low Resolution Image

Convolution

Decimation

ZSSR (Zero-Shot Super-Resolution)



Train on HR-LR pairs from the test image itself.

Totally unsupervised or self-supervised

Learns image specific internal information

Limitations

Requires thousands of gradient updates, which require a large amount of time.

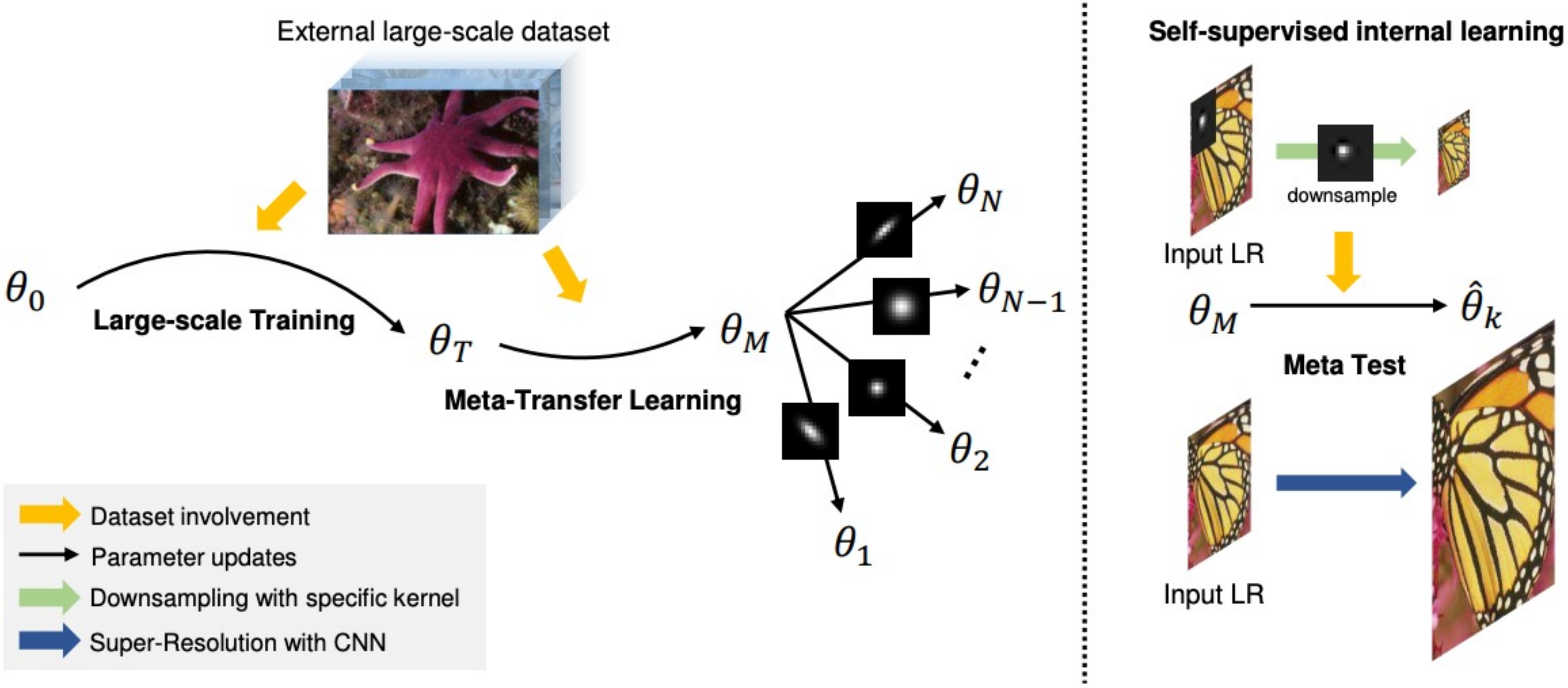
Cannot fully exploit large-scale external dataset.

Question 🤔

Proposed Methods

MZSR (Meta Transfer Learning for Zero Shot Super Resolution)

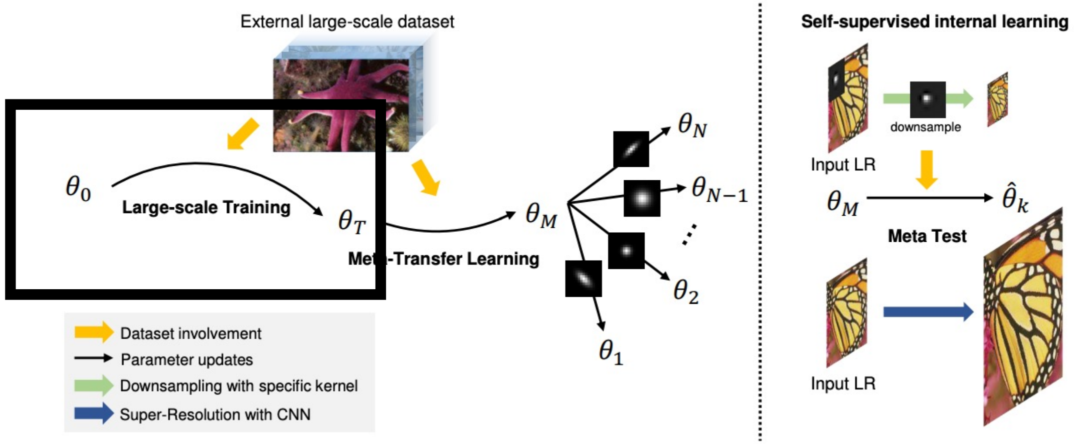
The overall scheme of our proposed MZSR



Large-scale Training

Synthesized large number of paired dataset.

Trained the network to learn super-resolution of "bicubic" degradation model by minimizing the loss.



Bicubic

$$\mathcal{L}^{\mathcal{D}}(\theta) = \mathbb{E}_{\mathcal{D}\sim(\mathbf{I}_{HR},\mathbf{I}_{LR}^{bic})}[||\mathbf{I}_{HR} - f_\theta(\mathbf{I}_{LR}^{bic})||_1]$$

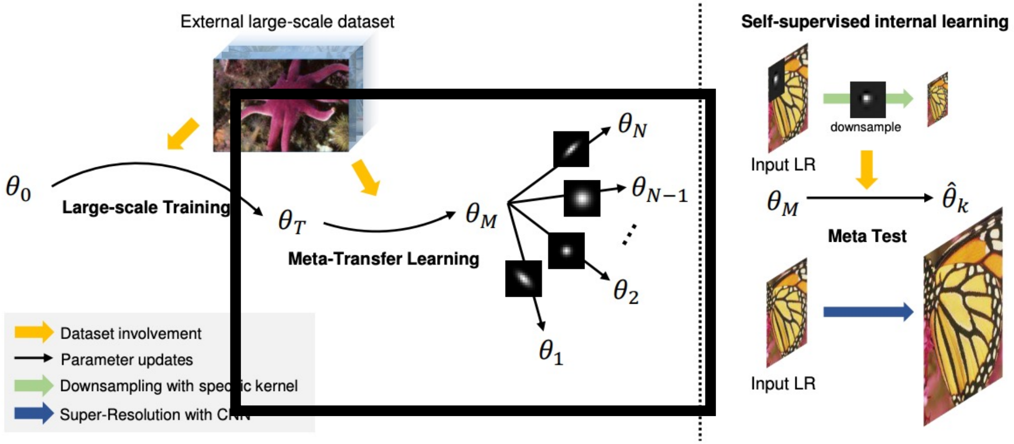
Ground-truth

Prediction

Paired dataset

Pixel-wise L1 loss

Meta-Transfer Learning



Adopt optimization-based meta-training step. Mostly follow MAML with some modifications.

Synthesize paired dataset with Gaussian kernels.

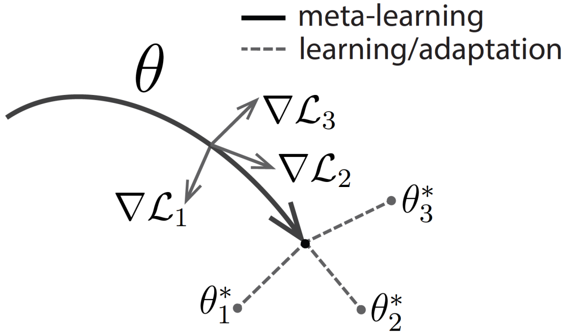
A kernel distribution is p(k) and each kernel is determined by a covariance matrix Σ.

Dataset for meta-transfer learning

$$D_{meta} : D_{tr}, D_{te}$$

Task-level training, test dataset

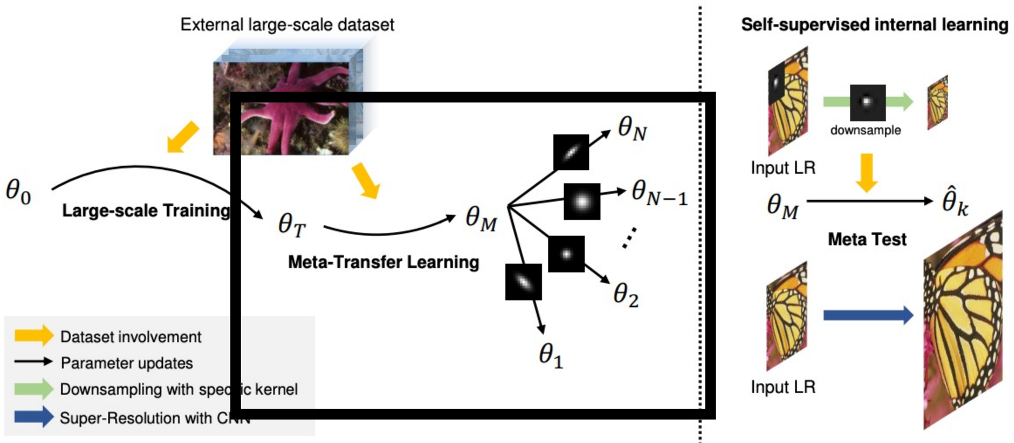
MAML(Model-Agnostic Meta-Learning)



Covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) \\ \sin(\Theta) & \cos(\Theta) \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \cos(\Theta) & \sin(\Theta) \\ -\sin(\Theta) & \cos(\Theta) \end{bmatrix}$$

Meta-Transfer Learning



Task-level gradient updates with the respect to model parameter θ.

The model parameters θ are optimized to achieve minimal test error of θ with respect to θ_i.

Parameter update

$$\theta_i = \theta - \alpha \nabla_\theta \mathcal{L}^{tr}_{\mathcal{T}_i}(\theta)$$

Task-level learning rate

Meta-transfer optimization

$$\arg\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}^{te}_{\mathcal{T}_i}(\theta - \alpha \nabla_\theta \mathcal{L}^{tr}_{\mathcal{T}_i}(\theta))$$

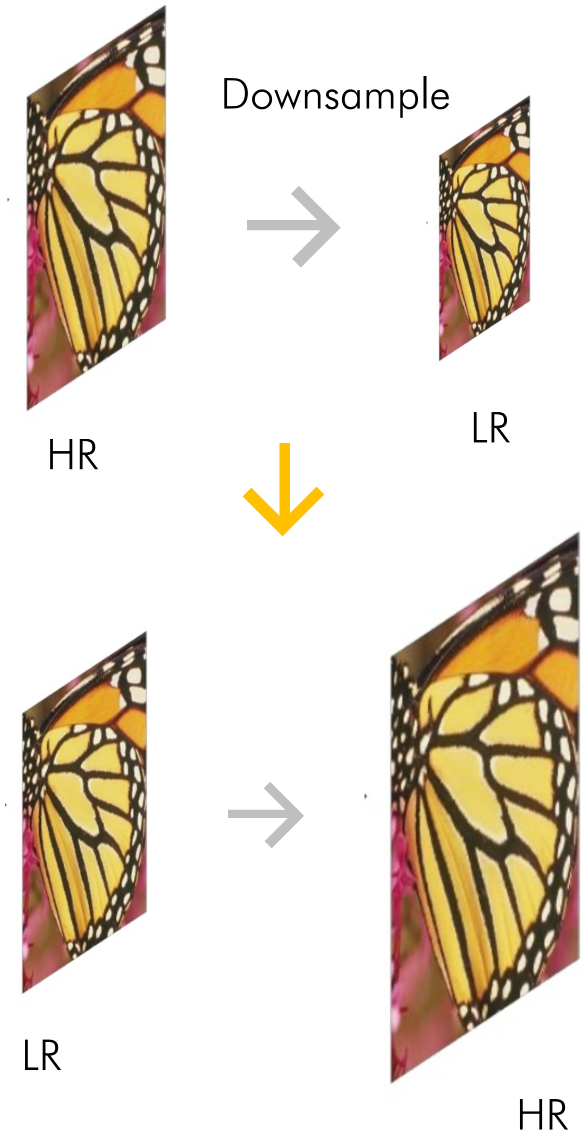
Kernel distribution

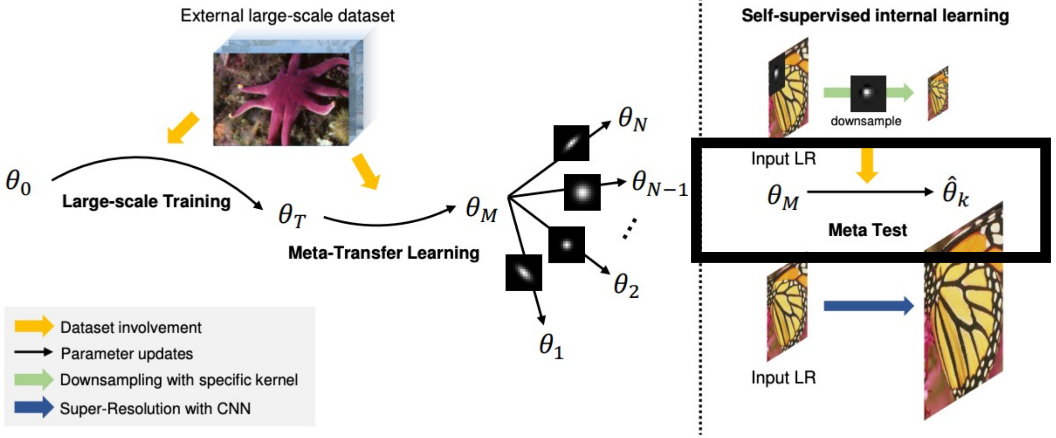
Parameter update rule

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}^{te}_{\mathcal{T}_i}(\theta_i),$$

Meta-learning rate

Meta-Test





Self-supervised ZSSR(Zero-Shot Super Resolution)

Learn internal information within a single image.

With a given LR image, we downsample it with corresponding downsampling kernel to generate $\mathbf{I}_{son}$(Image son)

Perform a few gradient updates with respect to the model parameter using a single pair of "LR son" and a given image.

Algorithm

Algorithm 1: Meta-Transfer Learning

Input: High-resolution dataset $\mathcal{D}_{\mathcal{HR}}$ and blur kernel distribution $p(\mathbf{k})$
Input: α, β: learning rates
Output: Model parameter θ_M

Randomly initialize θ
Synthesize paired dataset $\mathcal{D}$ by bicubicly downsample $\mathcal{D}_{HR}$

Large-scaling Training

while *not done* **do**
  Sample LR-HR batch from $\mathcal{D}$
  Evaluate $\mathcal{L}^D$ by Eq. 2
  Update θ with respect to $\mathcal{L}^D$
end

Generate task distribution $p(\mathcal{T})$ with $\mathcal{D}_{HR}$ and $p(\mathbf{k})$

Meta-transfer learning

while *not done* **do**
  Sample task batch $\mathcal{T}_i \sim p(\mathcal{T})$
  for *all* $\mathcal{T}_i$ **do**
    Evaluate training loss ($\mathcal{D}_{tr}$): $\mathcal{L}^{tr}_{\mathcal{T}_i}(\theta)$
    Compute adapted parameters with gradient descent: $\theta_i = \theta - \alpha \nabla_\theta \mathcal{L}^{tr}_{\mathcal{T}_i}(\theta)$
  end

Optimization

  Update θ with respect to average test loss ($\mathcal{D}_{te}$):
  $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}^{te}_{\mathcal{T}_i}(\theta_i)$
end

Algorithm 2: Meta-Test

Input: LR test image $\mathbf{I}_{LR}$, meta-transfer trained model parameter θ_M, number of gradient updates n and learning rate α
Output: Super-resolved image $\mathbf{I}_{SR}$

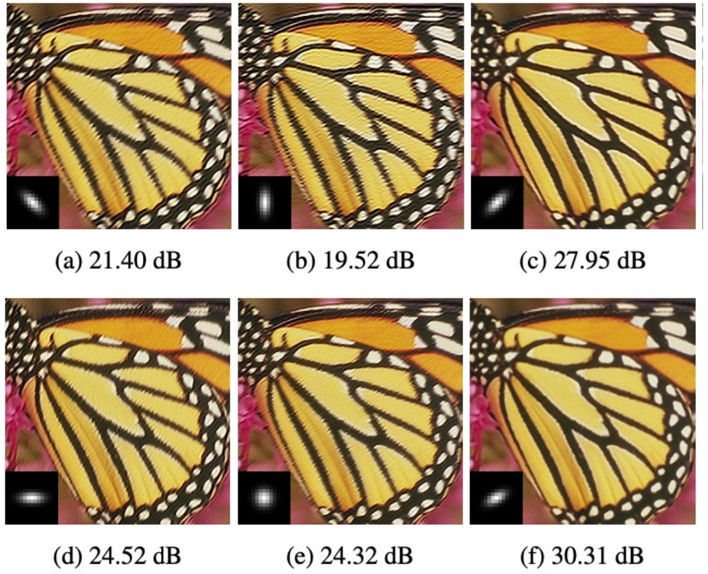
Initialize model parameter θ with θ_M
Generate LR son $\mathbf{I}_{son}$ by downsampling $\mathbf{I}_{LR}$ with corresponding blur kernel.
for *n steps* **do**
  Evaluate loss $\mathcal{L}(\theta) = ||\mathbf{I}_{LR} - f_\theta(\mathbf{I}_{son})||_1$
  Update $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta)$
end
return $\mathbf{I}_{SR} = f_\theta(\mathbf{I}_{LR})$

Question 🤔

Experiments

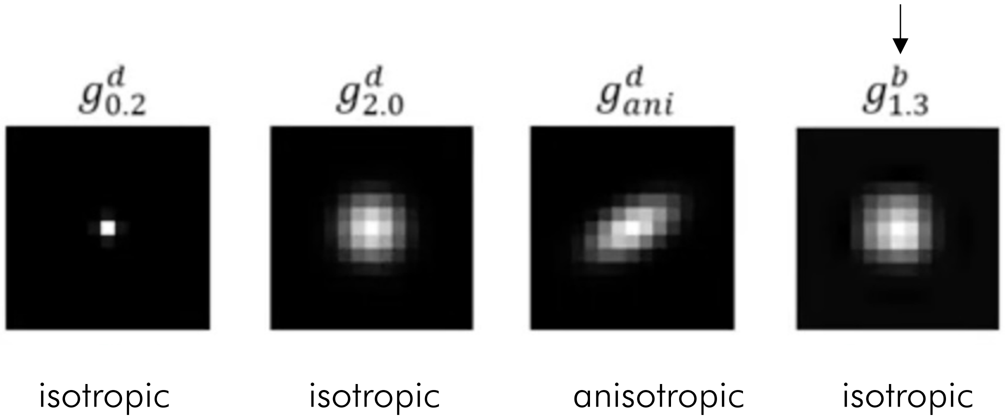
Measurement and Kernels

PSNR and SSIM



PSNR (dB): peak-signal-to-noise ratio
SSIM: structural index similarity

Gaussian blur kernels



d: direct subsampling
b: bicubic subsampling

Evaluations on "Bicubic" Downsampling

The average PSNR/SSIM results on "bicubic" downsampling scenario with ×2 on benchmarks

	Supervised			Unsupervised		
Dataset	Bicubic	CARN [2]	RCAN [45]	ZSSR [34]	MZSR (1)	MZSR (10)
Set5	33.64/0.9293	37.76/0.9590	38.18/0.9604	36.93/0.9554	36.77/0.9549	37.25/0.9567
BSD100	29.55/0.8427	32.09/0.8978	32.38/0.9018	31.43/0.8901	31.33/0.8910	31.64/0.8928
Urban100	26.87/0.8398	31.92/0.9256	33.30/0.9376	29.34/0.8941	30.01/0.9054	30.41/0.9092

Results on various kernel environments (X2)

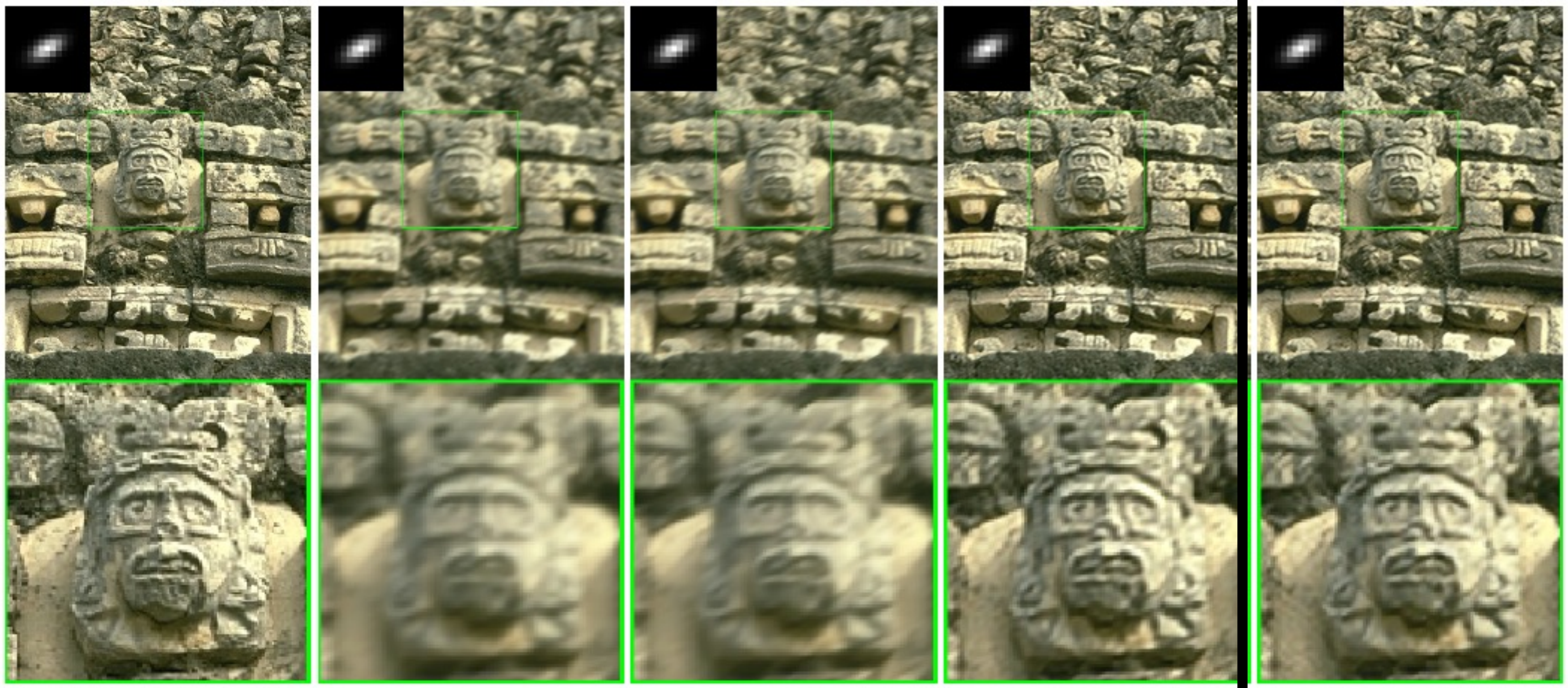
The average PSNR/SSIM results on various kernels with ×2 on benchmarks

		Supervised			Unsupervised		
Kernel	Dataset	Bicubic	RCAN [45]	IKC [11]	ZSSR [34]	MZSR (1)	MZSR (10)
$g^d_{0.2}$	Set5	30.24/0.8976	28.40/0.8618	29.09/0.8786	34.29/0.9373	33.14/0.9277	33.74/0.9301
	BSD100	27.45/0.7992	25.16/0.7602	26.23/0.7808	29.35/0.8465	28.74/0.8389	29.03/0.8415
	Urban100	24.70/0.7958	21.68/0.7323	23.66/0.7806	28.13/0.8788	26.24/0.8394	26.60/0.8439
$g^d_{2.0}$	Set5	28.73/0.8449	29.15/0.8601	29.05/0.8896	34.90/0.9397	35.20/0.9398	36.05/0.9439
	BSD100	26.51/0.7157	26.89/0.7394	27.46/0.8156	30.57/0.8712	30.58/0.8627	31.09/0.8739
	Urban100	23.70/0.7109	24.14/0.7384	25.17/0.8169	27.86/0.8582	28.23/0.8657	29.19/0.8838
g^d_{ani}	Set5	28.15/0.8265	28.42/0.8379	28.74/0.8565	33.96/0.9307	34.05/0.9271	34.78/0.9323
	BSD100	26.00/0.6891	26.22/0.7062	26.44/0.7310	29.72/0.8479	28.82/0.8013	29.54/0.8297
	Urban100	23.13/0.6796	23.35/0.6982	23.62/0.7239	27.03/0.8335	26.51/0.8126	27.34/0.8369
$g^b_{1.3}$	Set5	30.54/0.8773	31.54/0.8992	33.88/0.9357	35.24/0.9434	35.18/0.9430	36.64/0.9498
	BSD100	27.49/0.7546	28.27/0.7904	30.95/0.8860	30.74/0.8743	29.02/0.8544	31.25/0.8818
	Urban100	24.74/0.7527	25.65/0.7946	29.47/0.8956	28.30/0.8693	28.27/0.8771	29.83/0.8965

Visualized Results

Visualized comparisons of super-resolution results (×2)

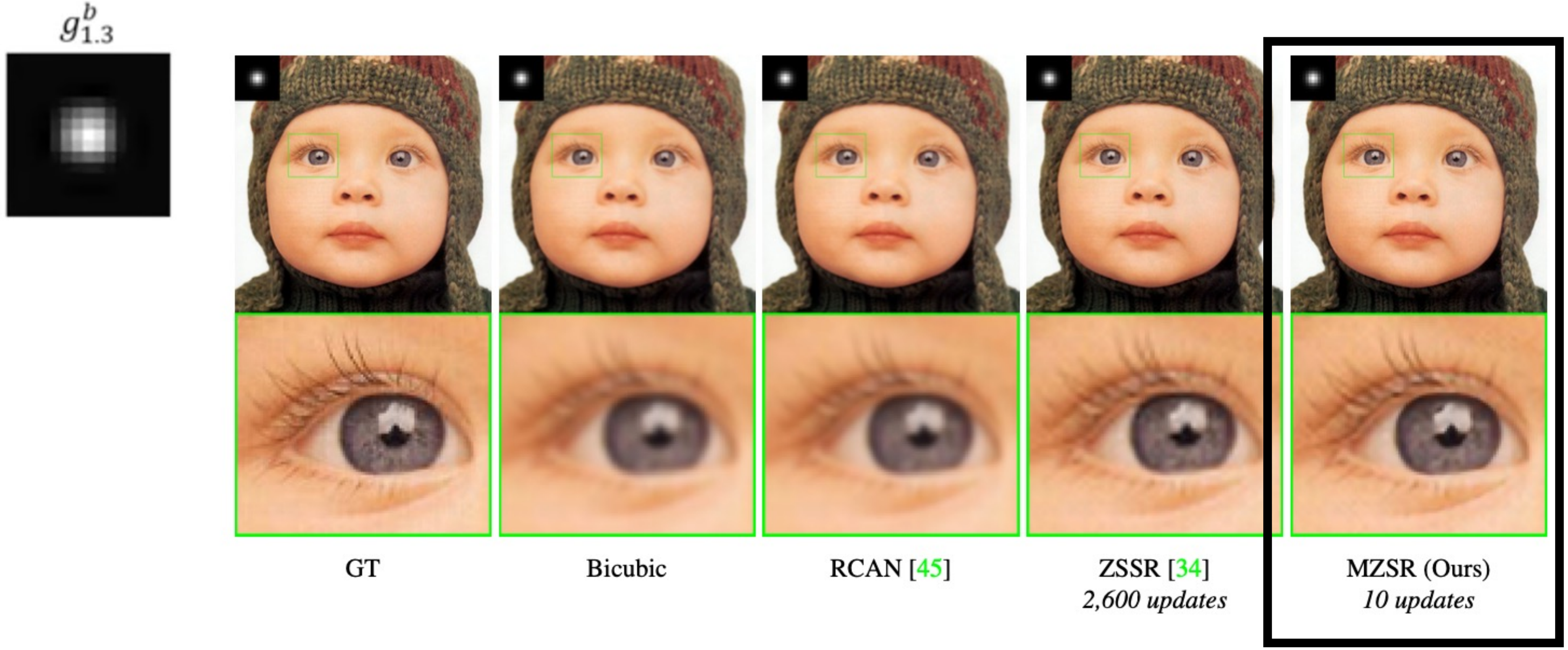
g^d_{ani}



GT | Bicubic | RCAN [4] | ZSSR [3] *2,160 updates* | MZSR (Ours) *10 updates*

Visualized Results

Visualized comparisons of super-resolution results (×2)

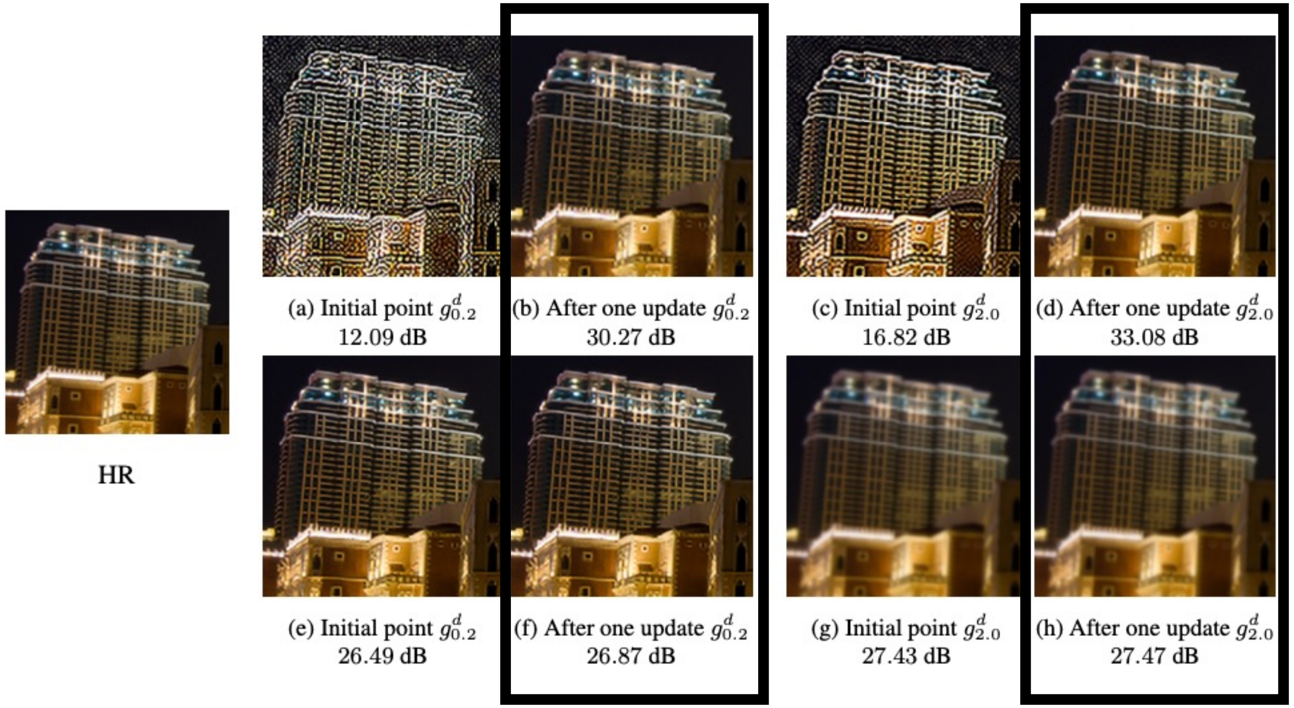


Discussion

Number of Gradient Updates

The initial point of MZSR shows the worst performance, but in one iteration, our method quickly adapts to the image condition and shows the best performance among the compared methods.

Visualization of the initial point and after one iteration of each method



MZSR

pre-trained network
on "bicubic" degradation

Multi-scale Models

With multiple scaling factors, the task distribution p(T) becomes more complex, in which the meta-learner struggles to capture such regions that are suitable for fast adaptation.

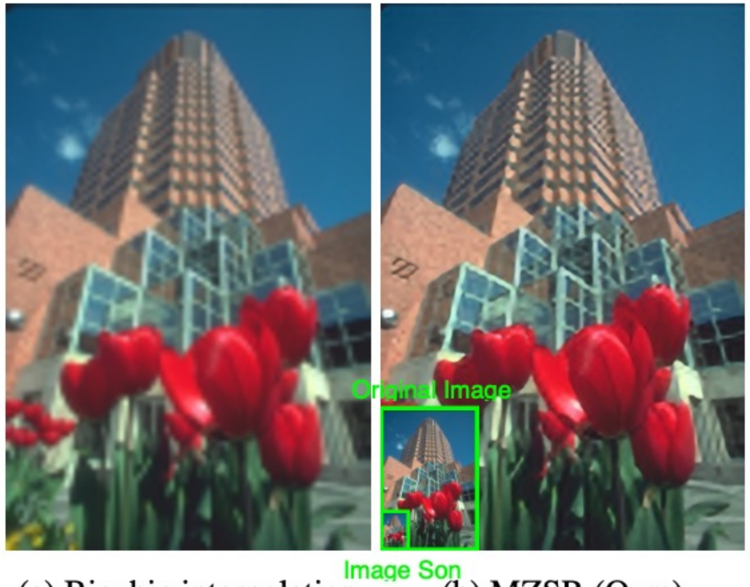
As MZSR learns internal information of CNN, such images with multi-scale recurrent patterns show plausible results even with large scaling factors.

MZSR results on scaling factor × 4 with blur kernel $g^d_{2.0}$

Average PSNR results of multi-scale model on Set5

PSNR (dB)	$g^d_{0.2}$	$g^d_{2.0}$	g^d_{ani}
Multi-scale (10)	33.33(−0.41)	35.67(−0.97)	33.95(−0.83)

The number in parenthesis is PSNR loss compared to the single-scale model.



(a) Bicubic interpolation (b) MZSR (Ours)

Complexity

MZSR with a single gradient update requires the shortest time among comparisons.

ZSSR requires thousands of forward and backward pass to get a super-resolved image.

Comparisons of the number of parameters and time complexity

Methods	Parameters	Time (sec)
CARN [2]	1,592 K	0.47
RCAN [45]	15,445 K	1.72
ZSSR [34]	225 K	142.72
MZSR (1)	225 K	0.13
MZSR (10)	225 K	0.36

Conclusion

Conclusion

Super-resolved results (X2)

(a) LR　(b) ZSSR [34] *2,850 updates*　(c) Fine-tuning *2,000 updates*　(d) MZSR (Ours) ***One** update*

MZSR is a fast, flexible, and lightweight self-supervised super-resolution method by exploiting both external and internal samples.

MZSR adopts an optimization-based meta-learning method with transfer learning to seek an initial point that is sensitive to different conditions of blur kernels.

MZSR can quickly adapt to specific image conditions within a few gradient updates.

MZSR outperforms other methods, including ZSSR, which requires thousands of gradient descent iterations.

Question 🤔

Sources

MZSR Paper (https://arxiv.org/pdf/2002.12213.pdf)
MZSR GitHub (https://github.com/JWSoh/MZSR)
ComputerVisionFoundation Videos (https://www.youtube.com/watch?v=ZEgZtvxuT4U)
Deep Learning Paper Review and Practice (https://www.youtube.com/watch?v=PUtFz4vqXHQ)

Presenter

Seonok Kim (sokim0991@korea.ac.kr)