Enhanced Deep Residual Networks *for* Single Image Super-Resolution





Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee

Computer Vision Lab.
Dept. of ECE, ASRI, Seoul National University
http://cv.snu.ac.kr

SISR (Single Image Super Resolution)

Goal: Restoring a HR image from a single LR image

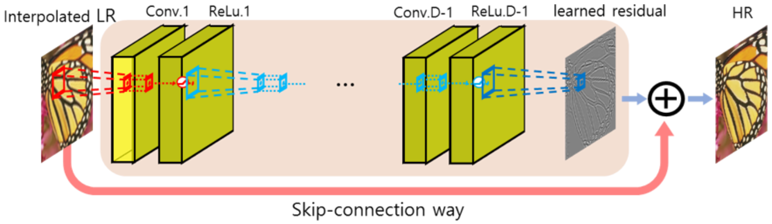
Super-Resolution

Low-resolution image

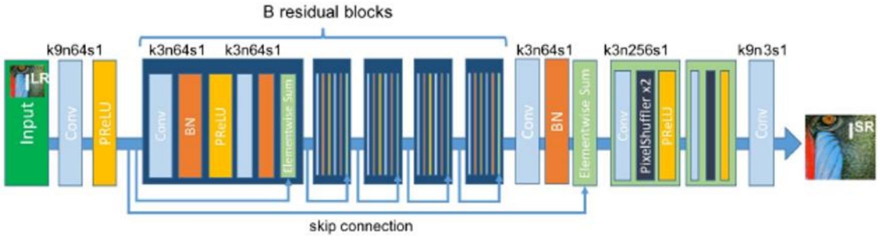
High-resolution image

ComputerVisionLab Seoul National University

Lessons from Recent Studies



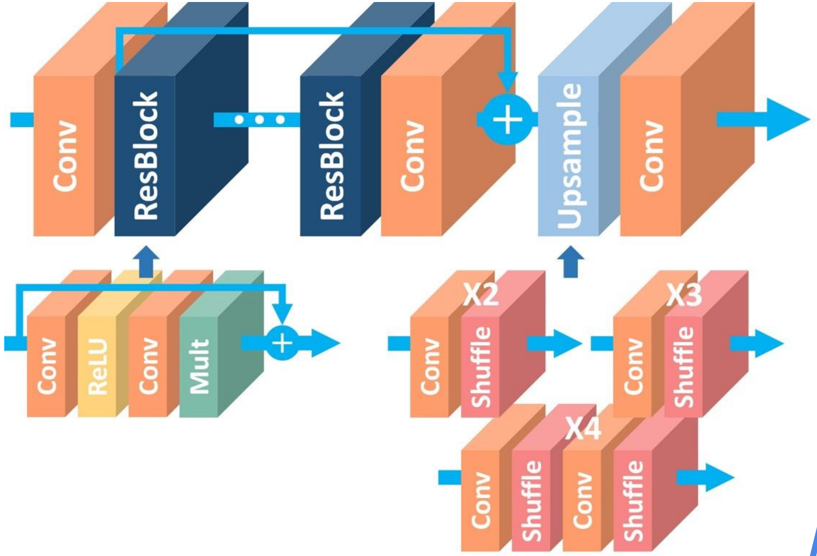
VDSR (CVPR2016)



SRResNet (CVPR2017)

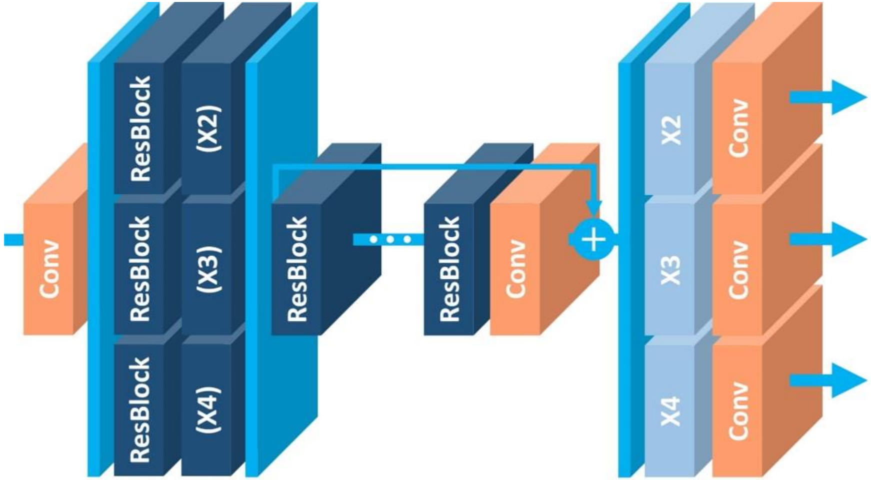
- *Skip connections*
  - Global and local skip connections enable deep architecture & stable training
- *Upscaling methods*
  - <u>Post-upscaling</u> using sub-pixel convolution is more efficient than <u>pre-upscaling</u>
  - However, they are limited that only single-scale SR is possible

EDSR



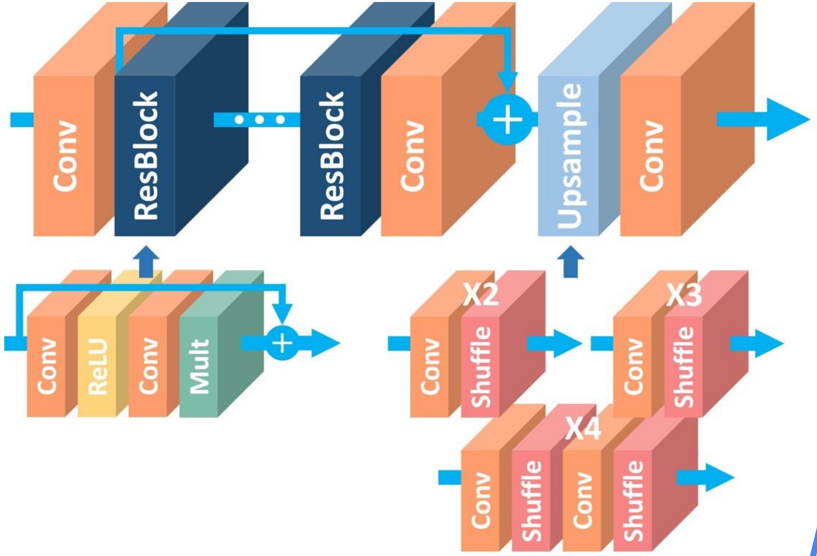
Conv
ResBlock
ResBlock
Conv
Upsample
Conv
Conv
ReLU
Conv
Mult
X2
Conv
Shuffle
X3
Conv
Shuffle
X4
Conv
Shuffle
Conv
Shuffle

MDSR



Conv
ResBlock
(X2)
ResBlock
(X3)
ResBlock
(X4)
ResBlock
ResBlock
Conv
X2
Conv
X3
Conv
X4
Conv

EDSR



4 Techniques for Better SR

Need Batch-Normalization?

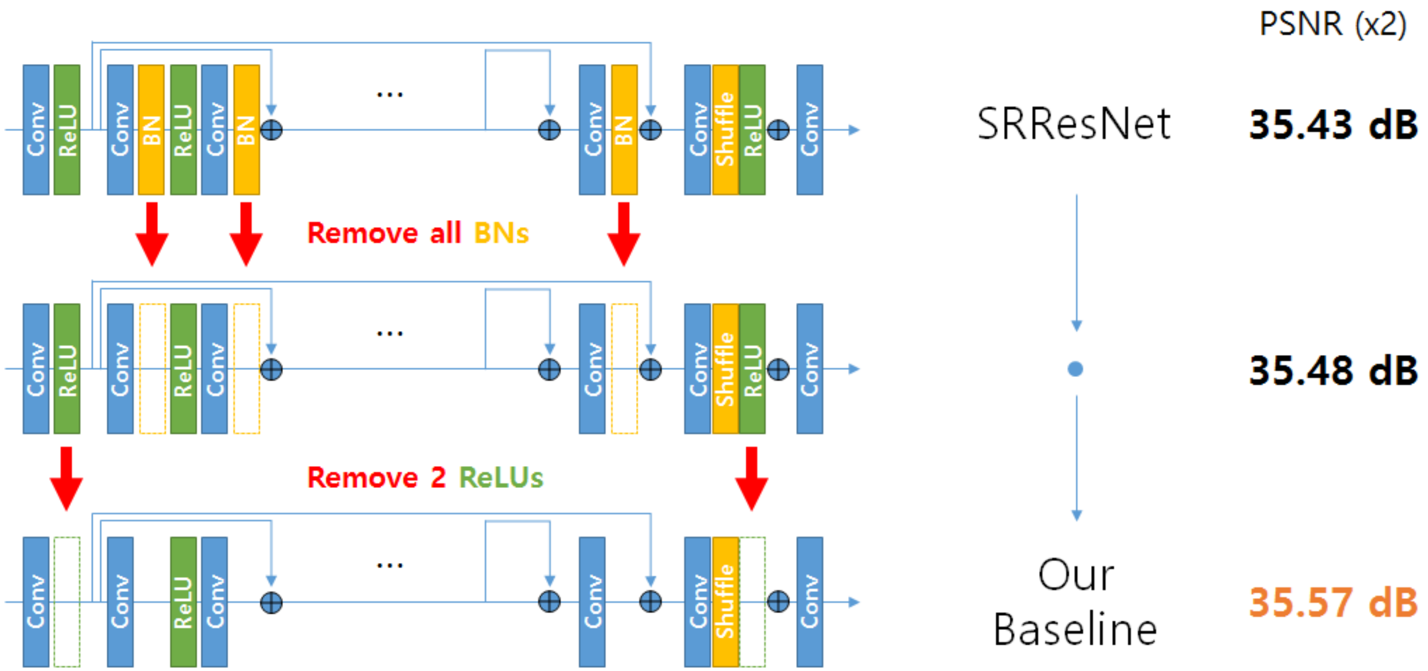
Increasing model size

Better loss function

Geometric self-ensemble

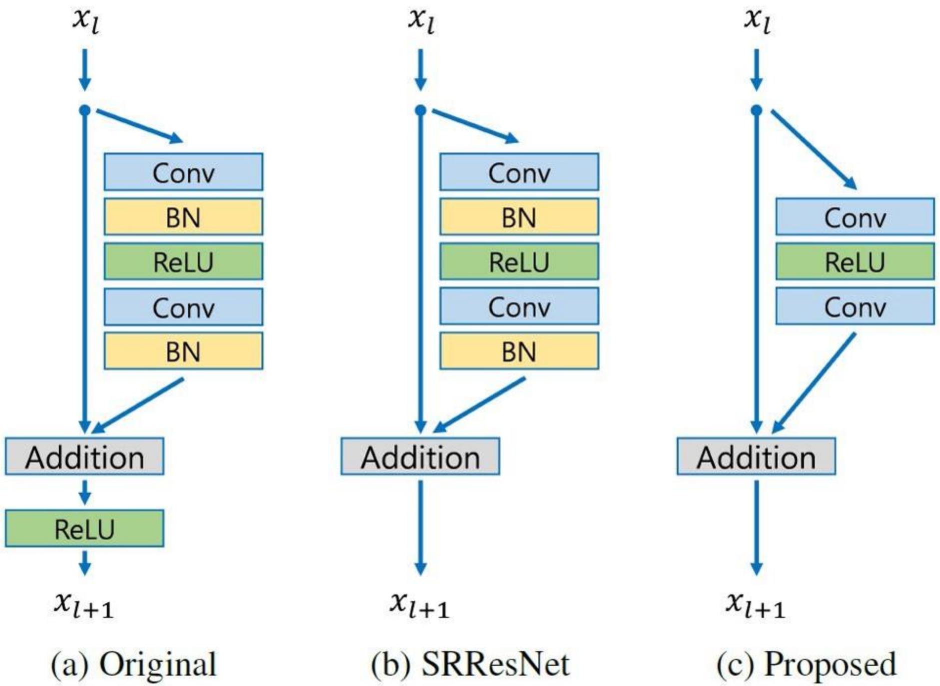
Need Batch-Normalization?

Empirical tests show that removing Batch-Normalization improves the performance!



Need Batch-Normalization?

- Unlike classification problem, input and output have similar distributions
- In SR, normalizing intermediate features may not be desirable
- Also, can save ~40% of memory
  → Can enlarge the model size

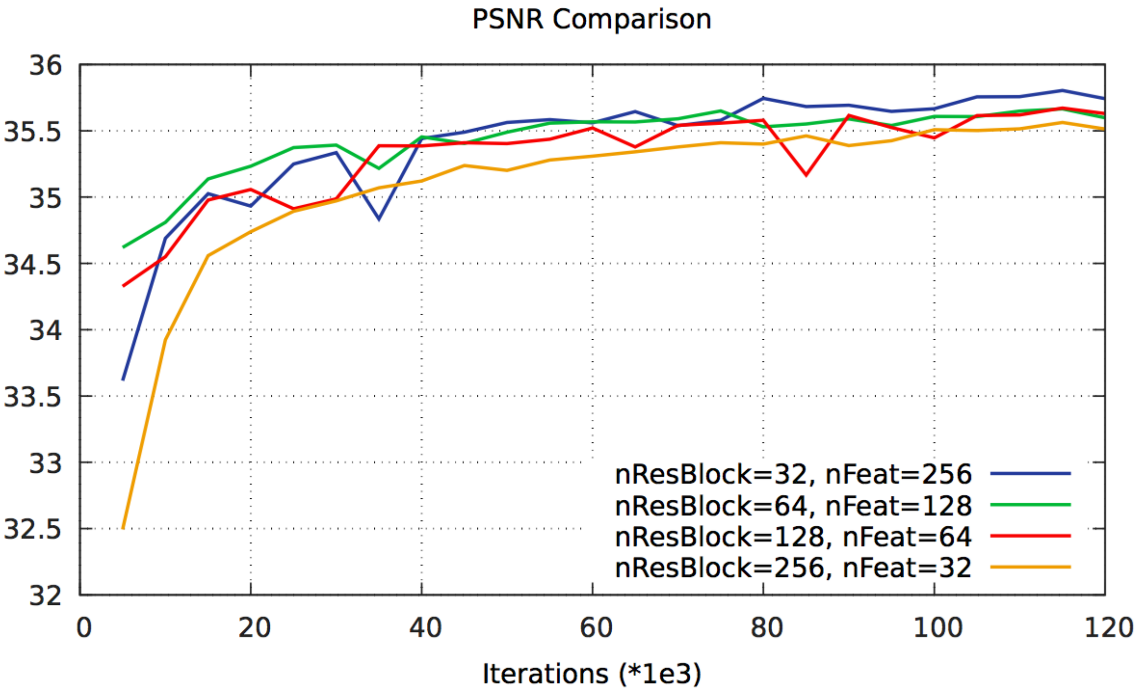


Increasing Model Size

Given a limited memory, which design is better?

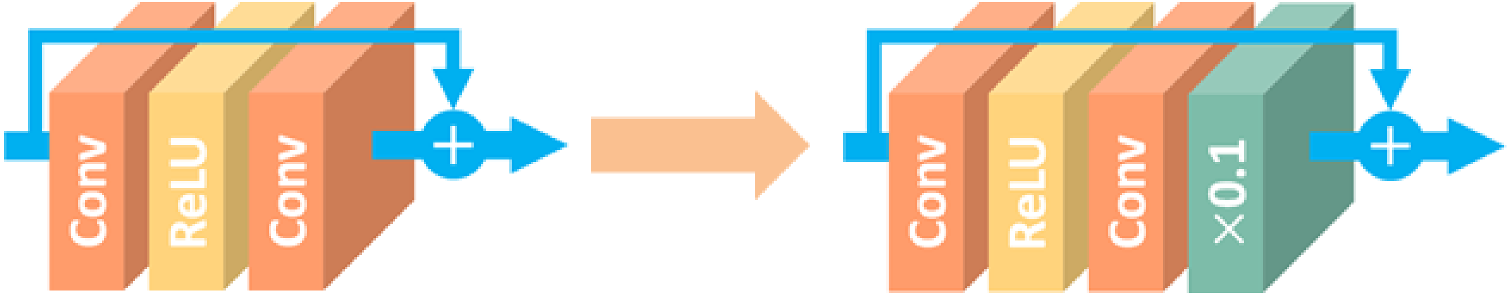
- Empirical test show that increasing *#features* is better than increasing depth

- Instability occurs when #features increased up to 256



Increasing Model Size

- Residual Scaling Layer
  - Increasing #features (up to 256) results instability during training
  - Constant scaling layers after each residual path prevents such instability



Proposed in (Szegedy 2016), "*Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*"

Loss Function: L1 vs L2

- Is MSE (L2 loss) the best choice?
- Comparison between different loss functions
    - EDSR baseline(16 res-blocks), scale=2, tested on DIV2K images (791~800)

Loss	Definition	PSNR
l_2 (MSE)	$\frac{1}{N}\sum_{i=1}^{N}\left\|I_i - \tilde{I}_i\right\|_2^2$	35.46 dB
l_1 (MAE)	$\frac{1}{N}\sum_{i=1}^{N}\left\|I_i - \tilde{I}_i\right\|_1$	35.55 dB

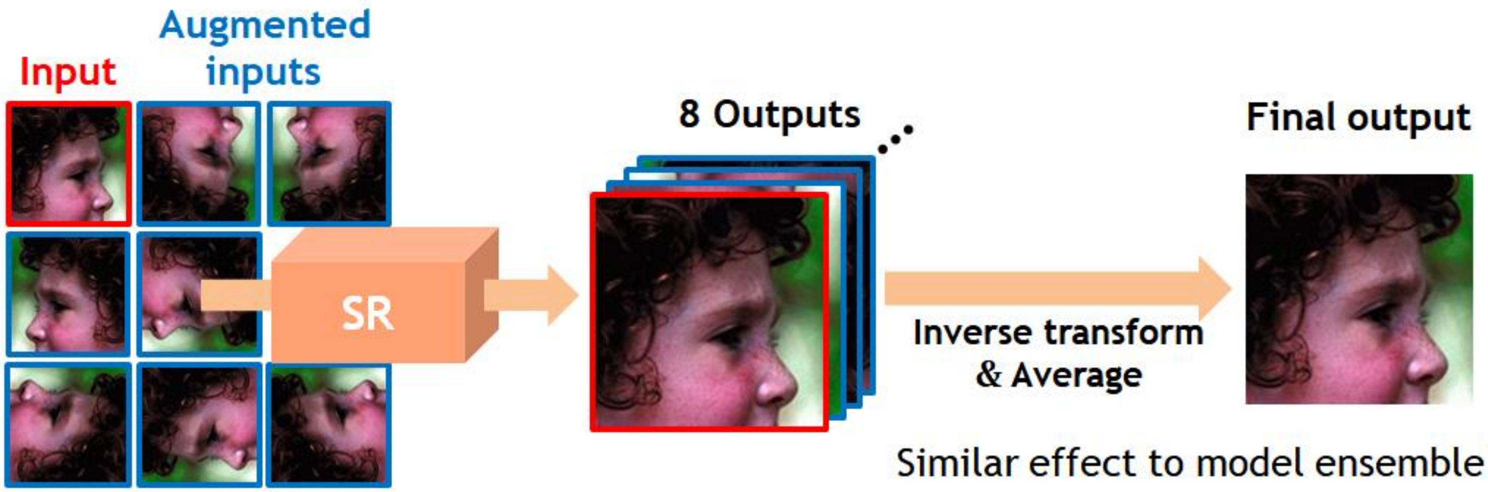
→ *MSE is not a good choice!*

Geometric Self-Ensemble

- Motivation
  - Model ensemble is nice, but expensive!
  - How can we achieve an ensemble effect while avoiding training new models?
- Method
  - Transform test image 8 times with flips and rotations (x8)
  - Build 8 outputs and inverse-transform correspondingly
  - Average 8 results

Proposed in (Timofte 2016), "*Seven ways to improve example-based single-image super-resolution*"

Geometric Self-Ensemble

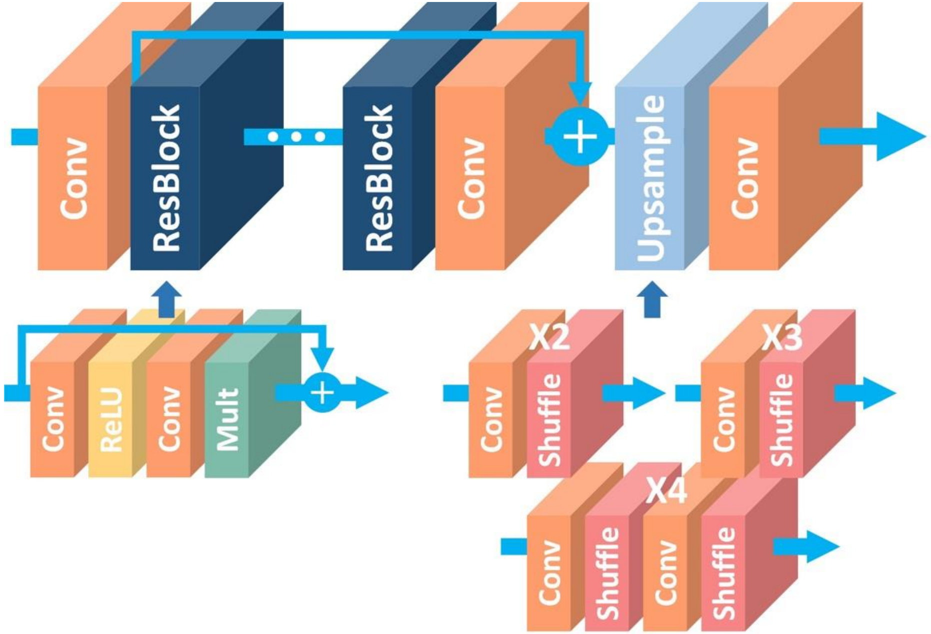


Similar effect to model ensemble!

Augmentation	X (×1)	Vertical Flip (×2)	Flip (×4)	Flip + Rotation (×8)
PSNR(dB)	35.55	35.61 (+ 0.06)	35.65 (+ 0.10)	**35.67 (+ 0.12)**

EDSR Summary

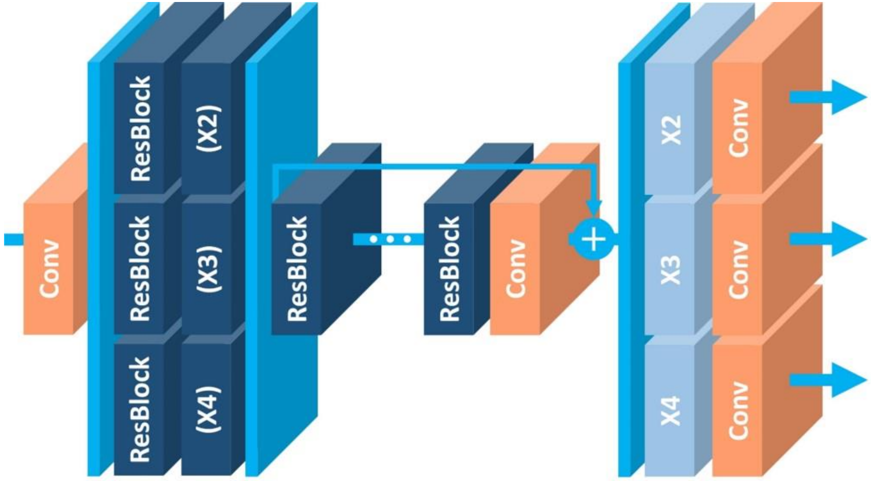
- Deeper & Wider: 32 ResBlocks and 256 channels
- Global-local skip connections
- Post-upscaling
- No Batch-Normalization
- Residual scaling
- L1 loss function
- Geometric self-ensemble (EDSR+)



EDSR

Conv
ResBlock
ResBlock
Conv
Upsample
Conv
Conv
ReLU
Conv
Mult
X2
Conv
Shuffle
X3
Conv
Shuffle
X4
Conv
Shuffle
Conv
Shuffle

MDSR



Conv
ResBlock
(X2)
ResBlock
(X3)
ResBlock
(X4)
ResBlock
ResBlock
Conv
X2
Conv
X3
Conv
X4
Conv

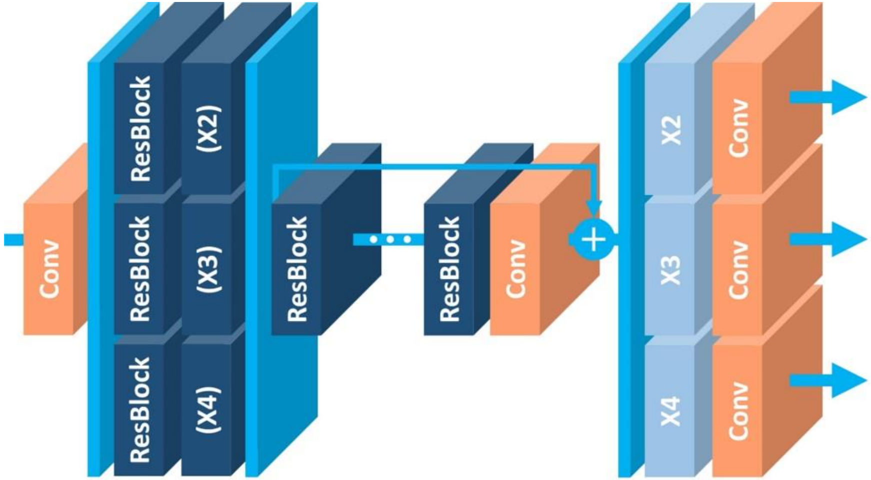
Motivation

- VDSR: Multi-scale SR in a single model
- Multi-scale knowledge transfer

Efficient Multi-Scale Model

- Designing MDSR
- Single vs. Multi-scale learning
- Train & Test method
- EDSR vs. MDSR

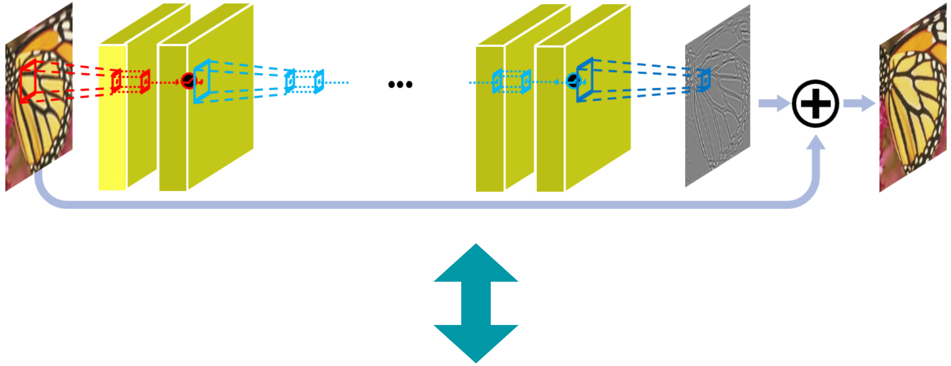
MDSR



Motivation

SRCNN, VDSR: A single architecture regardless of upscaling factor

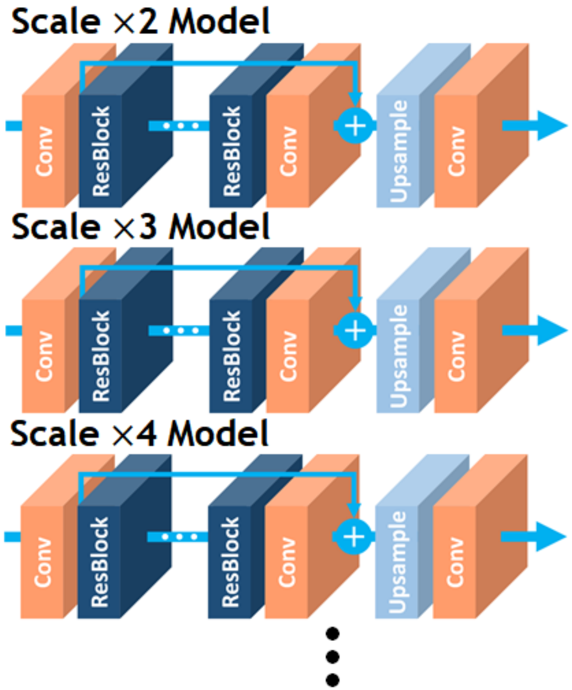
⇨ Multi-scale SR in a single model (*VDSR*)



FSRCNN, ESPCN, SRResNet: Fast & Efficient, (*late upsampling*)

but cannot deal with the multiple scales in a single model.

Motivation

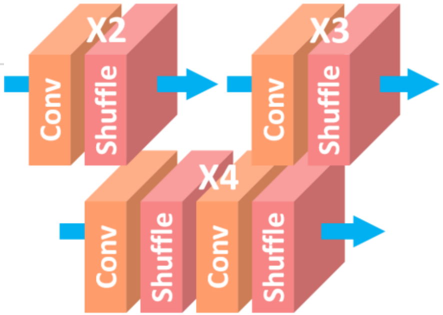


FSRCNN, ESPCN, SRResNet

⇨ Different models for different scales?

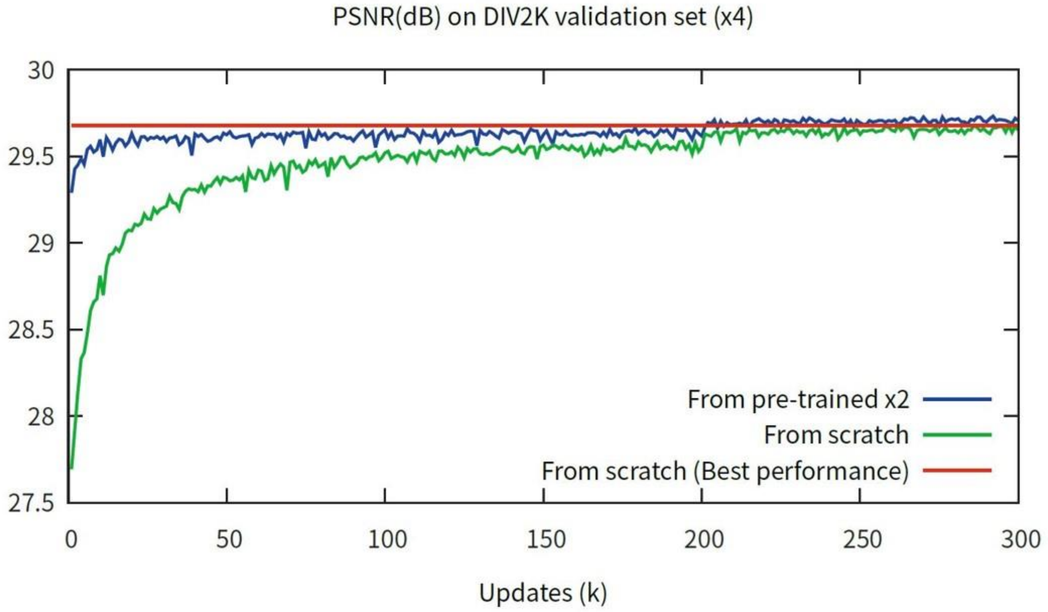
- Heavy training burden
- Waste of parameters for similar tasks
- Redundancy

Motivation



Multi-scale knowledge transfer

- Pre-trained scale x2 networks greatly helps training scale x3 and x4 networks.
- Super-resolution at multiple scales are inter-related tasks!



Designing MDSR

How to make *EDSR* (post-upscaling) to handle multiscale SR as VDSR?

Requirements

1. Reduce the variance between the different scales

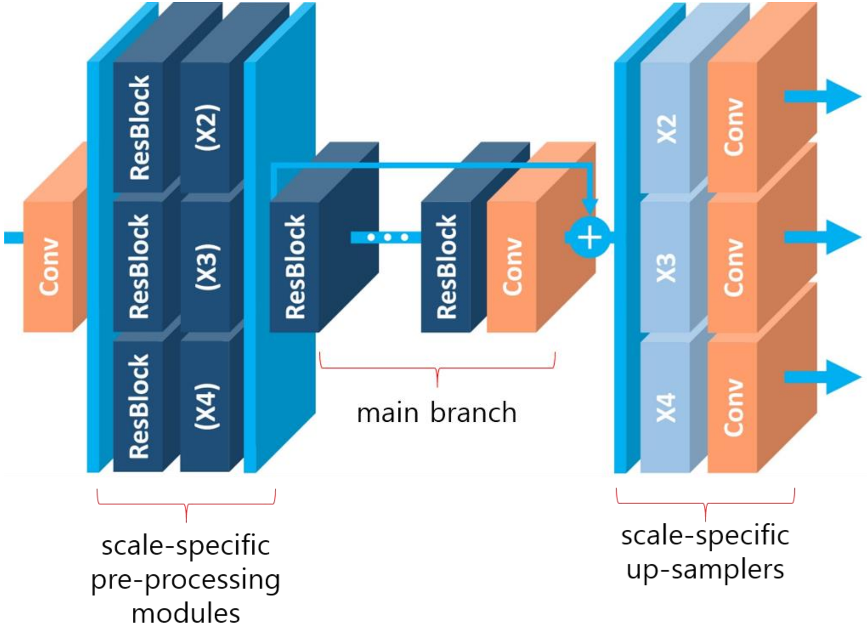
   ⇨ *Scale-specific pre-processing modules*

2. Most parameters are shared across scales

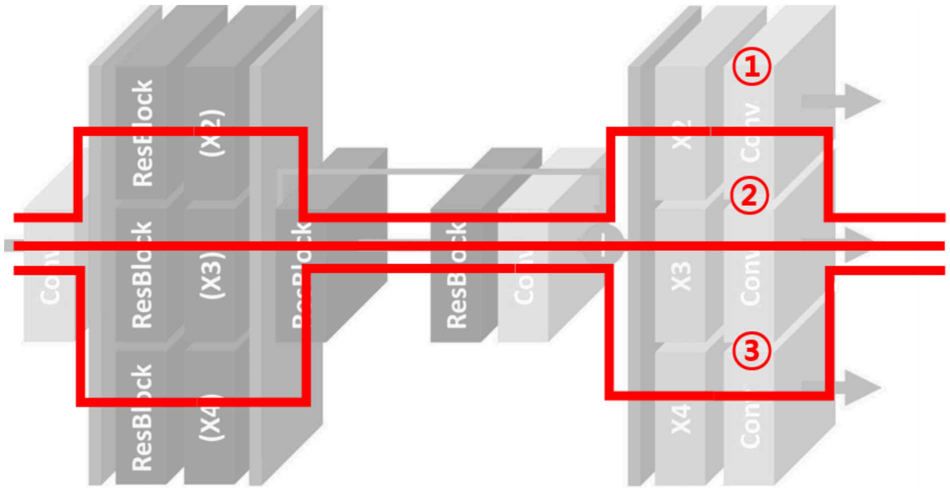
   ⇨ *main branch*

3. For efficiency: Post-upscaling

   ⇨ *Scale-specific up-samplers*



Train and Test Method



1. *Train*

   - Only one of 3 scale-specific branches is activated at each iteration
   - A mini-batch consists of single-scale patches

2. *Test*

   - Select one of the paths (①~③) according to the desired SR scale

EDSR vs. MDSR

- *Performance:*
  MDSR ≲ EDSR

Scale	SRResNet (L2 loss)	SRResNet (L1 loss)	Our baseline (Single-scale)	Our baseline (Multi-scale)	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
×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
×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

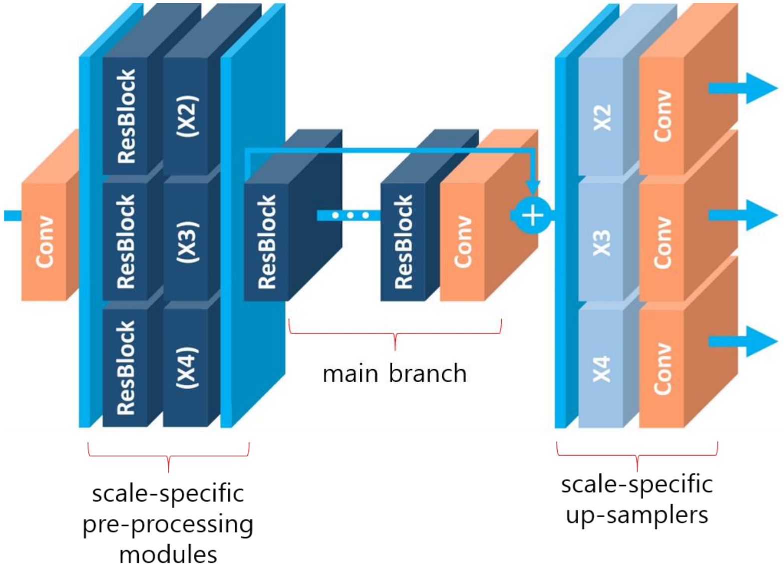
- *# Parameters:*
  MDSR << EDSR
  (Almost ⅕! + MDSR can handle the multiple scales in a single model)

- *Stability:*
  MDSR << EDSR
  (We failed to increase #features even with residual scaling)

Options	SRResNet [14] (reproduced)	Baseline (Single / Multi)	EDSR	MDSR
# Residual blocks	16	16	32	80
# Features	64	64	256	64
# Parameters	1.5M	1.5M / 3.2M	43M	8.0M
Residual scaling	-	-	0.1	-
Use BN	Yes	No	No	No
Loss function	L2	L1	L1	L1

MDSR Summary

- Very deep architecture: 80 ResBlocks
- Most parameters are shared in main branch
- Scale-specific pre-processing modules and up-samplers
- Post-upscaling
- No Batch-Normalization
- L1 loss function
- Geometric self-ensemble (MDSR+)



Results

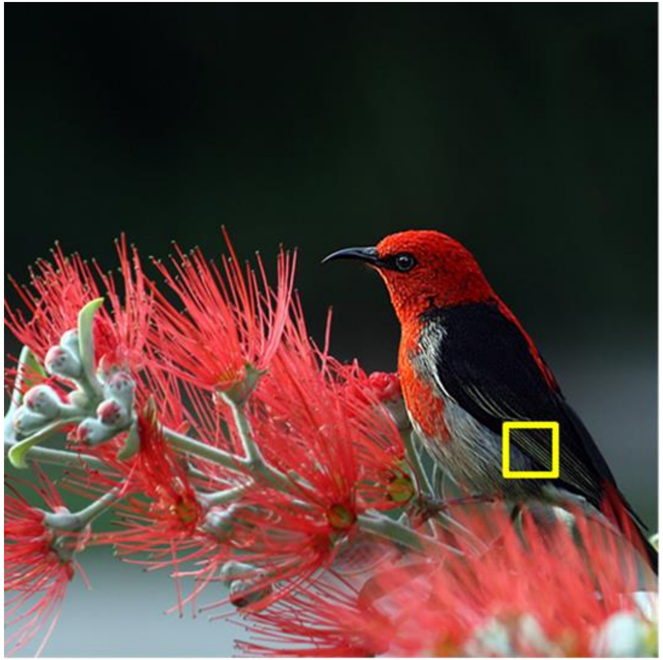
Training Details

Item	Detail
Input channels	3 (RGB)
# Training images	800
Patch size	96×96 (×2), 144×144 (×3), 192×192 (×4),
Mini-batch size	16
Learning rate	10^{-4}, halved at every 2×10^5 iterations
Data augmentation	Flips and Rotations (×8)
Optimizer	ADAM ($\beta_1 = 0.9$)
Loss	l_1
# Iteration	6×10^5 (4 days in single Titan X)
Implementation	Torch7 (Lua)
Dataset	DIV2K

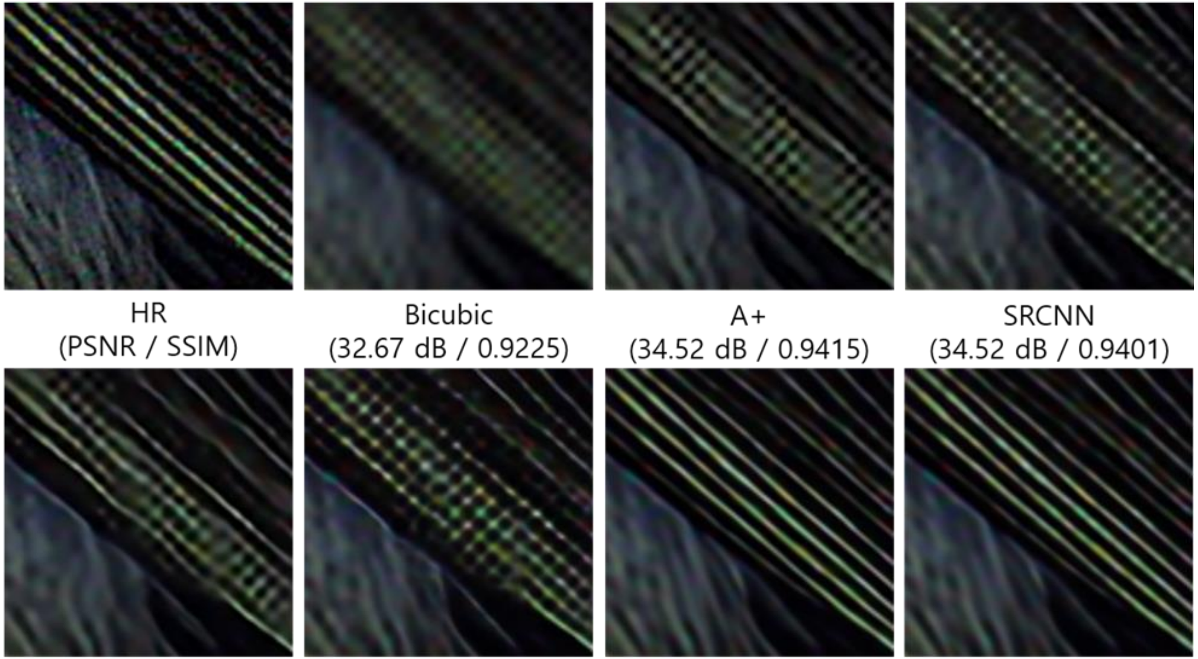
Quantitative Results

Dataset	Scale	Bicubic	A+ [27]	SRCNN [4]	VDSR [11]	SRResNet [14]	EDSR (Ours)	MDSR (Ours)	**EDSR+ (Ours)**	**MDSR+ (Ours)**
Set5	×2	33.66 / 0.9299	36.54 / 0.9544	36.66 / 0.9542	37.53 / 0.9587	- / -	38.11 / 0.9601	38.11 / 0.9602	38.20 / 0.9606	38.17 / 0.9605
	×3	30.39 / 0.8682	32.58 / 0.9088	32.75 / 0.9090	33.66 / 0.9213	- / -	34.65 / 0.9282	34.66 / 0.9280	34.76 / 0.9290	34.77 / 0.9288
	×4	28.42 / 0.8104	30.28 / 0.8603	30.48 / 0.8628	31.35 / 0.8838	32.05 / 0.8910	32.46 / 0.8968	32.50 / 0.8973	32.62 / 0.8984	32.60 / 0.8982
Set14	×2	30.24 / 0.8688	32.28 / 0.9056	32.42 / 0.9063	33.03 / 0.9124	- / -	33.92 / 0.9195	33.85 / 0.9198	34.02 / 0.9204	33.92 / 0.9203
	×3	27.55 / 0.7742	29.13 / 0.8188	29.28 / 0.8209	29.77 / 0.8314	- / -	30.52 / 0.8462	30.44 / 0.8452	30.66 / 0.8481	30.53 / 0.8465
	×4	26.00 / 0.7027	27.32 / 0.7491	27.49 / 0.7503	28.01 / 0.7674	28.53 / 0.7804	28.80 / 0.7876	28.72 / 0.7857	28.94 / 0.7901	28.82 / 0.7876
B100	×2	29.56 / 0.8431	31.21 / 0.8863	31.36 / 0.8879	31.90 / 0.8960	- / -	32.32 / 0.9013	32.29 / 0.9007	32.37 / 0.9018	32.34 / 0.9014
	×3	27.21 / 0.7385	28.29 / 0.7835	28.41 / 0.7863	28.82 / 0.7976	- / -	29.25 / 0.8093	29.25 / 0.8091	29.32 / 0.8104	29.30 / 0.8101
	×4	25.96 / 0.6675	26.82 / 0.7087	26.90 / 0.7101	27.29 / 0.7251	27.57 / 0.7354	27.71 / 0.7420	27.72 / 0.7418	27.79 / 0.7437	27.78 / 0.7425
Urban100	×2	26.88 / 0.8403	29.20 / 0.8938	29.50 / 0.8946	30.76 / 0.9140	- / -	32.93 / 0.9351	32.84 / 0.9347	33.10 / 0.9363	33.03 / 0.9362
	×3	24.46 / 0.7349	26.03 / 0.7973	26.24 / 0.7989	27.14 / 0.8279	- / -	28.80 / 0.8653	28.79 / 0.8655	29.02 / 0.8685	28.99 / 0.8683
	×4	23.14 / 0.6577	24.32 / 0.7183	24.52 / 0.7221	25.18 / 0.7524	26.07 / 0.7839	26.64 / 0.8033	26.67 / 0.8041	26.86 / 0.8080	26.86 / 0.8082
DIV2K validation	×2	31.01 / 0.9393	32.89 / 0.9570	33.05 / 0.9581	33.66 / 0.9625	- / -	35.03 / 0.9695	34.96 / 0.9692	35.12 / 0.9699	35.05 / 0.9696
	×3	28.22 / 0.8906	29.50 / 0.9116	29.64 / 0.9138	30.09 / 0.9208	- / -	31.26 / 0.9340	31.25 / 0.9338	31.39 / 0.9351	31.36 / 0.9346
	×4	26.66 / 0.8521	27.70 / 0.8736	27.78 / 0.8753	28.17 / 0.8841	- / -	29.25 / 0.9017	29.26 / 0.9016	29.38 / 0.9032	29.36 / 0.9029

Qualitative Results



'0853' from DIV2K



HR
(PSNR / SSIM)

Bicubic
(32.67 dB / 0.9225)

A+
(34.52 dB / 0.9415)

SRCNN
(34.52 dB / 0.9401)

VDSR
(35.17 dB / 0.9463)

SRResNet
(35.95 dB / 0.9496)

EDSR+ (Ours)
(36.74 dB / 0.9544)

MDSR+ (Ours)
(36.74 dB / 0.9543)

Qualitative Results

‘img039’ from Urban100

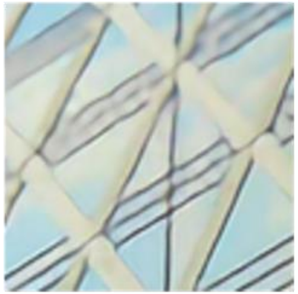
HR
(PSNR / SSIM)

Bicubic
(20.85 dB / 0.5844)

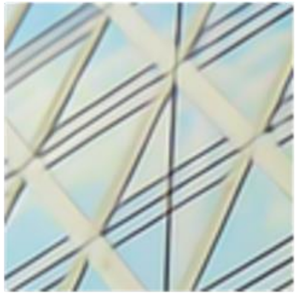
A+
(21.58 dB / 0.6443)

SRCNN
(21.69 dB / 0.6510)

VDSR
(22.25 dB / 0.6918)



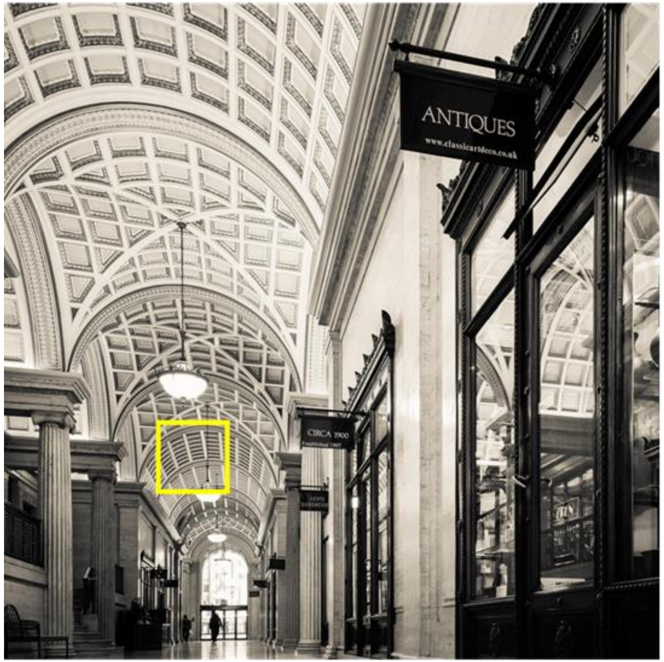
SRResNet
(23.28 dB / 0.7568)



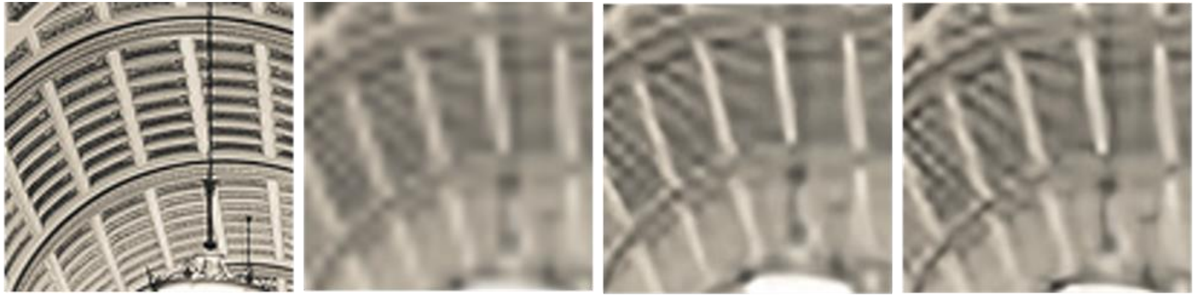
EDSR+ (Ours)
(23.98 dB / 0.7911)

MDSR+ (Ours)
(23.89 dB / 0.7881)

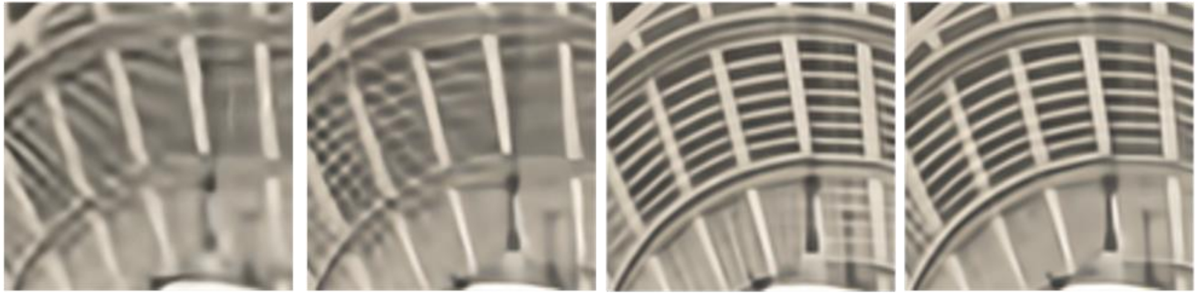
Qualitative Results



'img083' from Urban100

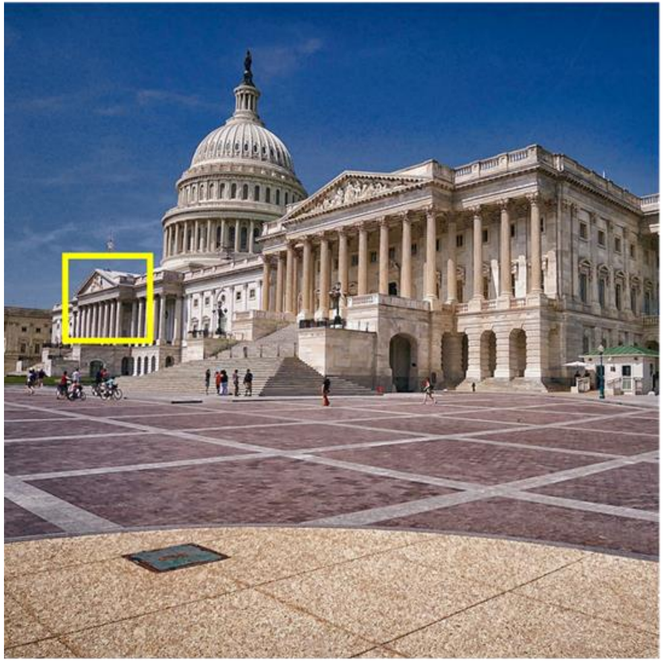


HR (PSNR / SSIM) | Bicubic (20.46 dB / 0.5544) | A+ (21.27 dB / 0.6235) | SRCNN (21.35 dB / 0.6284)



VDSR (21.73 dB / 0.6632) | SRResNet (22.33 dB / 0.7005) | **EDSR+ (Ours)** (**22.86 dB / 0.7369**) | **MDSR+ (Ours)** (**22.90 dB / 0.7363**)

Qualitative Results



'img070' from Urban100

HR
(PSNR / SSIM)

Bicubic
(21.48 dB / 0.5263)

A+
(21.80 dB / 0.5642)

SRCNN
(21.82 dB / 0.5646)

VDSR
(21.91 dB / 0.5773)

SRResNet
(22.16 dB / 0.5953)

EDSR+ (Ours)
(22.39 dB / 0.6122)

MDSR+ (Ours)
(22.37 dB / 0.6106)

Qualitative Results

'0841' from DIV2K

HR
(PSNR / SSIM)

Bicubic
(27.61 dB / 0.8115)

A+
(28.48 dB / 0.8449)

SRCNN
(28.57 dB / 0.8445)

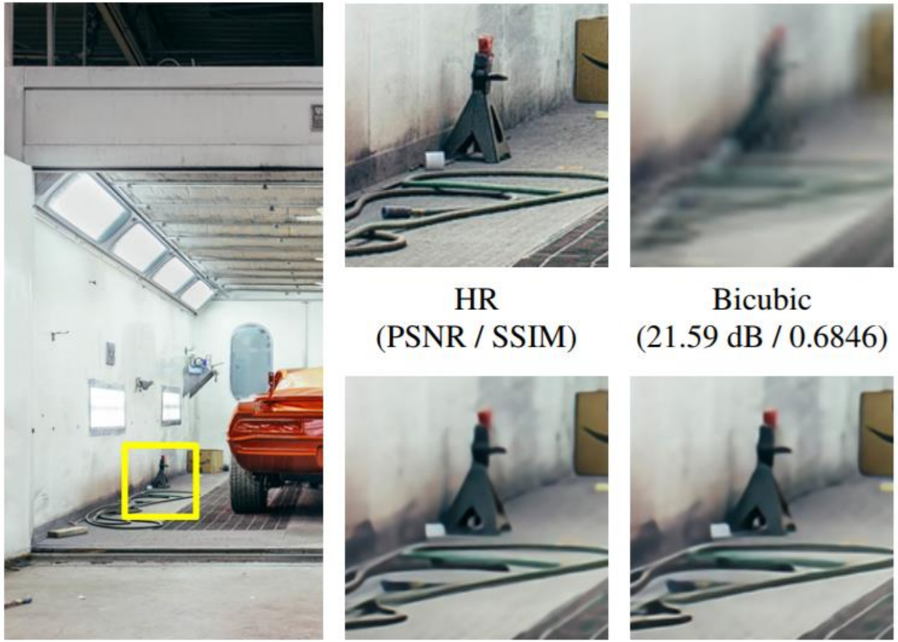
VDSR
(28.93 dB / 0.8589)

SRResNet
(29.44 dB / 0.8727)

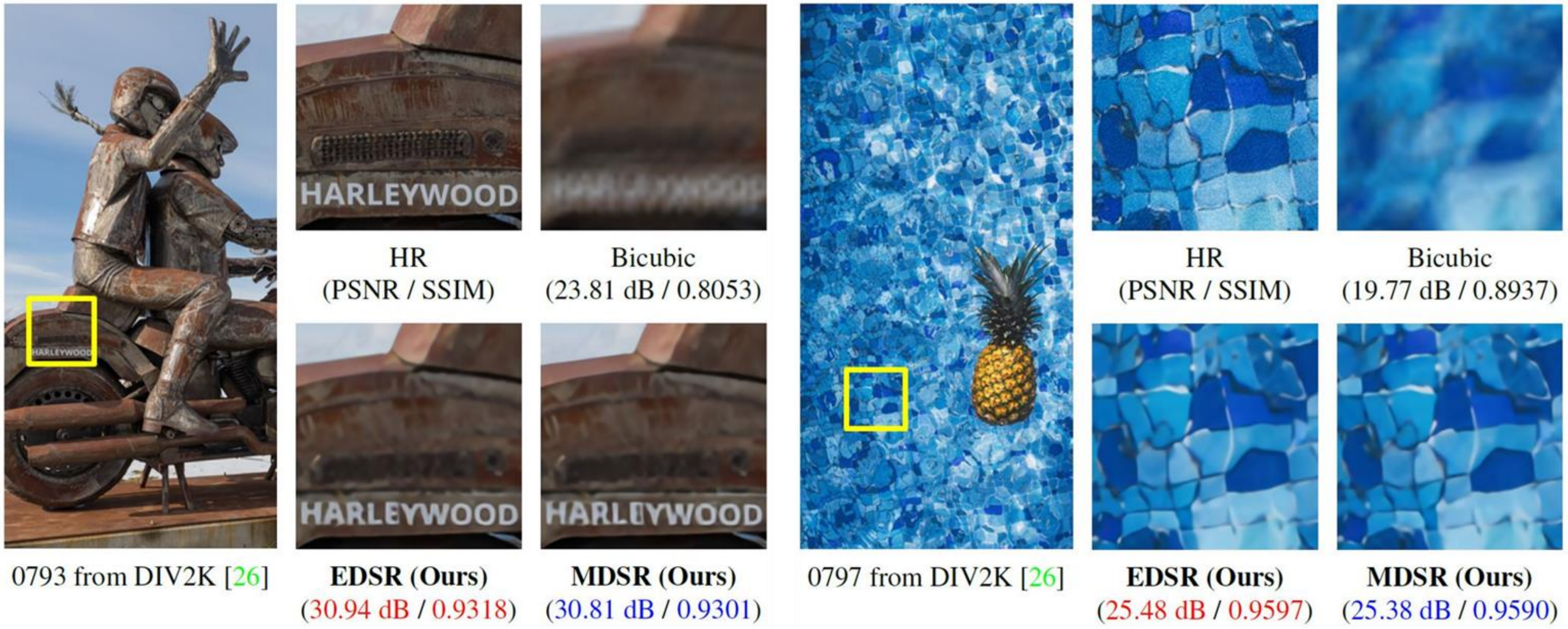
EDSR+ (Ours)
(30.07 dB / 0.8864)

MDSR+ (Ours)
(30.03 dB / 0.8848)

Unknown Track (Challenge)



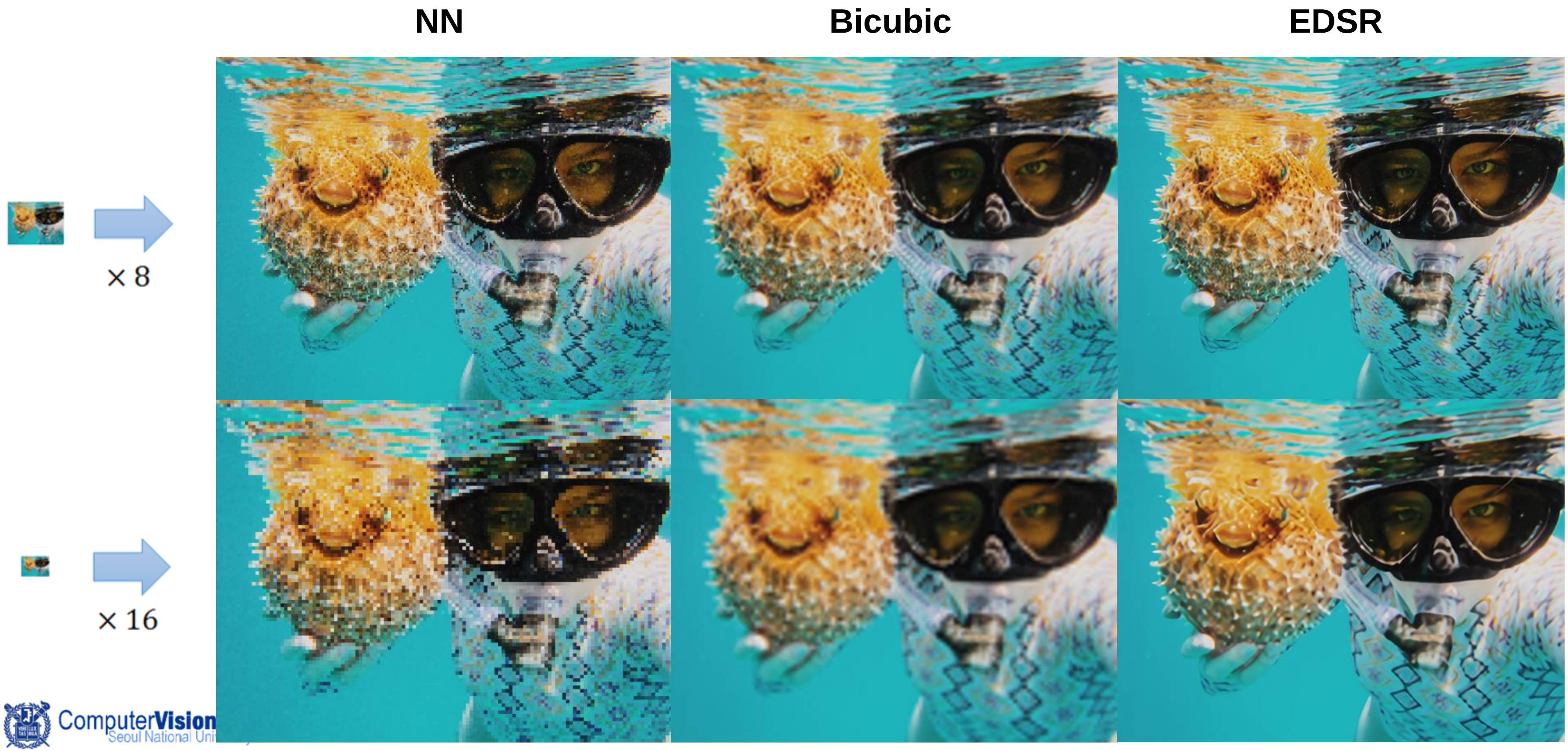
Unknown Track (Challenge)



Extreme SR (up to x64)

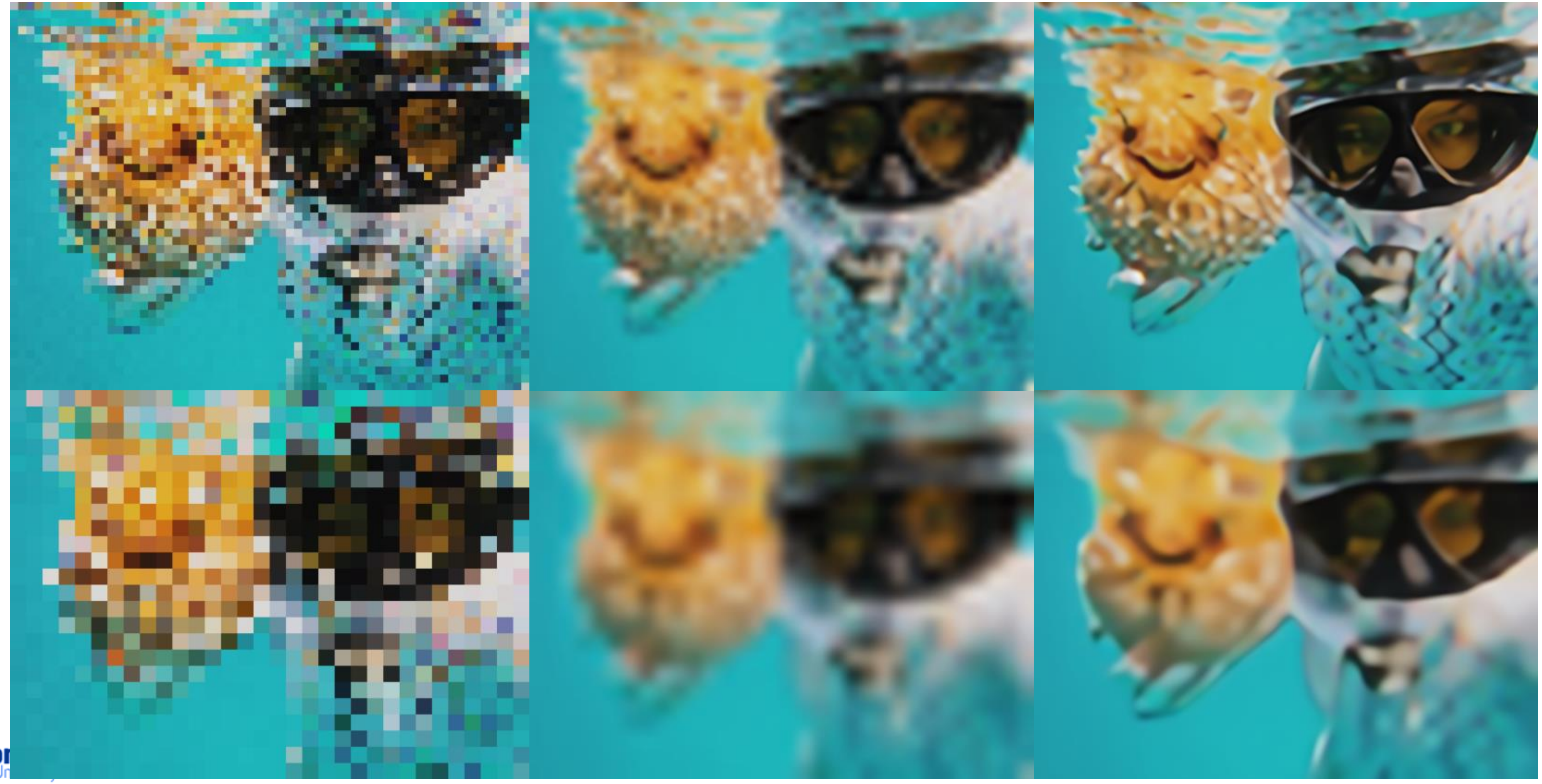
How about extreme cases?

Extreme SR (up to x64)



ComputerVision
Seoul National Uni

Extreme SR (up to x64)



Conclusion

1. *State-of-the-art single image super-resolution system using better ResNet structure*

2. *Techniques to build & train extremely large model*

3. *A single network to deal with multi-scale SR problem*

Thank you!

http://cv.snu.ac.kr