IVCL

Enhanced Deep Residual Networks for Single Image Super-Resolution

Bee Lim et al., 2017

NTIRE 2017 winner, CVPRW 2017 best paper

Presenter: Le Van The

Outline

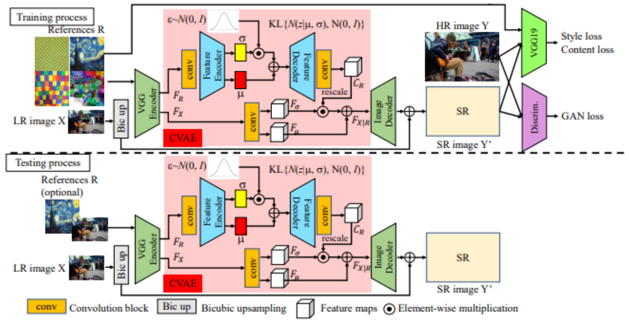
- NTIRE challenge on Single Image Super-Resolution
- Proposed EDSR and MDSR
- Experiment results
- Conclusion

NTIRE challenge on Image Super-Resolution

NTIRE (New Trends in Image Restoration and Enhancement)

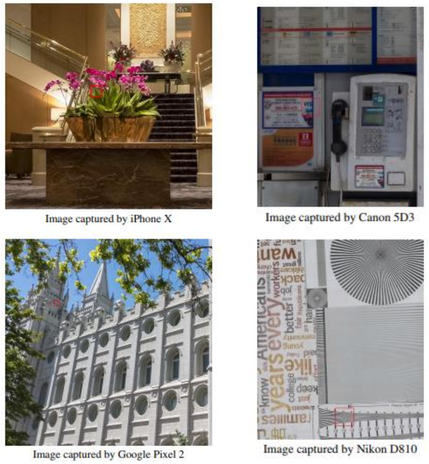
- NTIRE 2017 Challenge on single image Super-Resolution
- NTIRE 2018 Challenge on single image Super-Resolution
- NTIRE 2019 Challenge on Real Image Super-Resolution
- NTIRE 2020 Challenge on real-world image Super-Resolution
- NTIRE 2021 Learning the Super-Resolution Space Challenge

✓ Gauge and push the state-of-the-art in SR
✓ Compare different SR solutions
✓ Promote new dataset benchmark



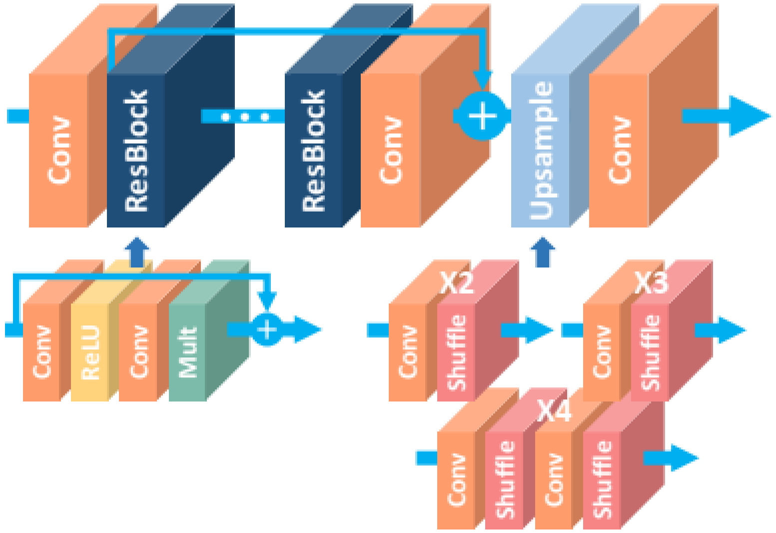
VAE method of SR_DL team at NTIRE 2021

Qualitative comparison at NTIRE 2020

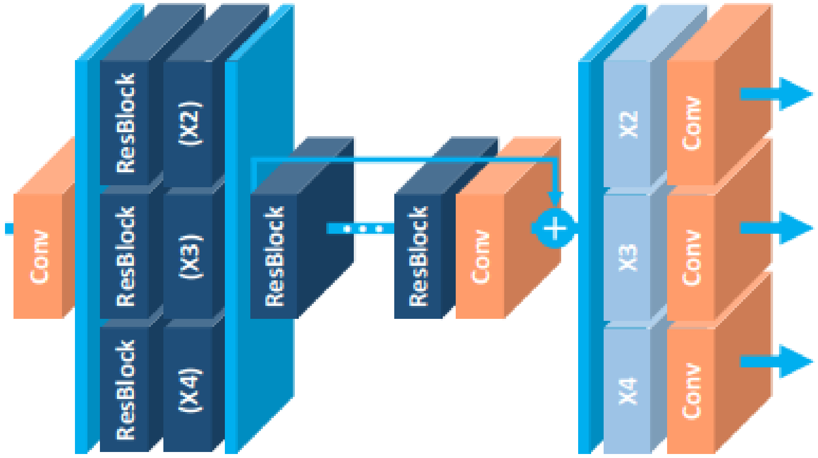


RealSR dataset at NTIRE 2020

Proposed model

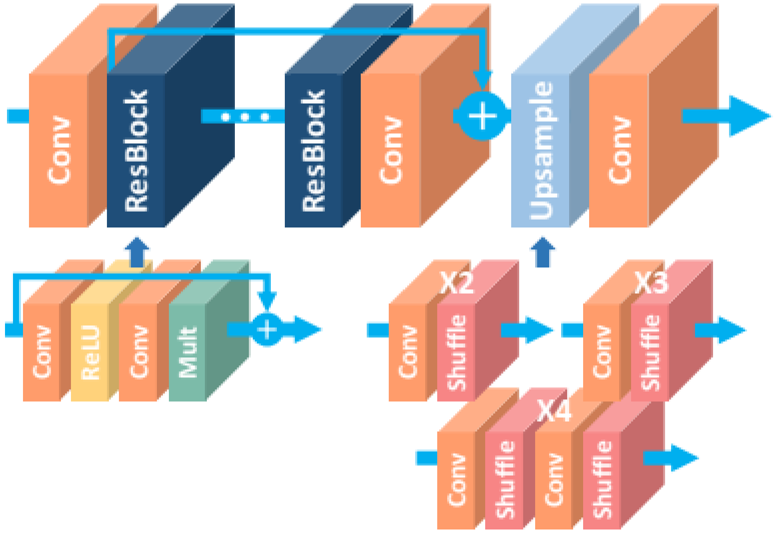


Single-scale SR network (EDSR)



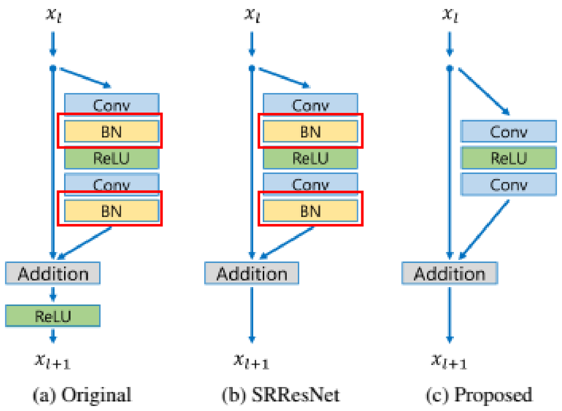
Multi-scale SR network (MDSR)

EDSR



Single-scale SR network (EDSR)

❖ Improve residual block: Remove batch normalization (BN) layer

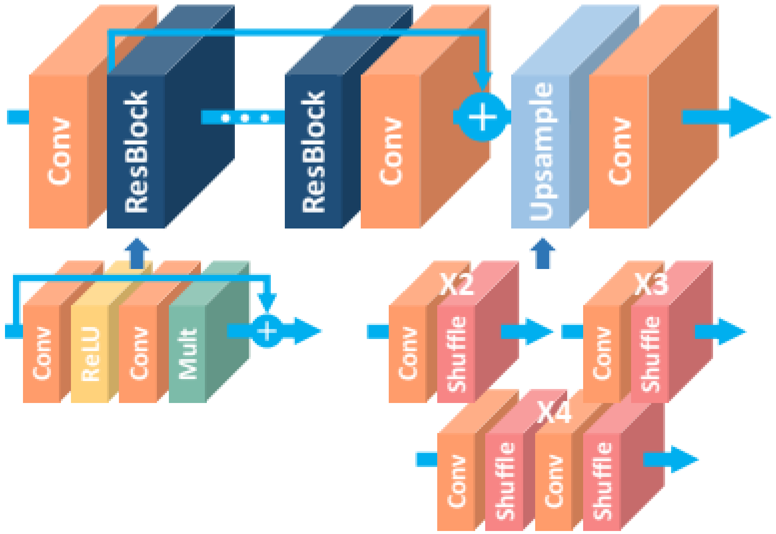


Comparison of residual blocks

- Motivation :
  - input and output have same distribution
  - BN layers get rid of range flexibility
  - BN make model heavier
- Effect:
  - saves approximately 40% of memory usage
  - overcome limited computational resource

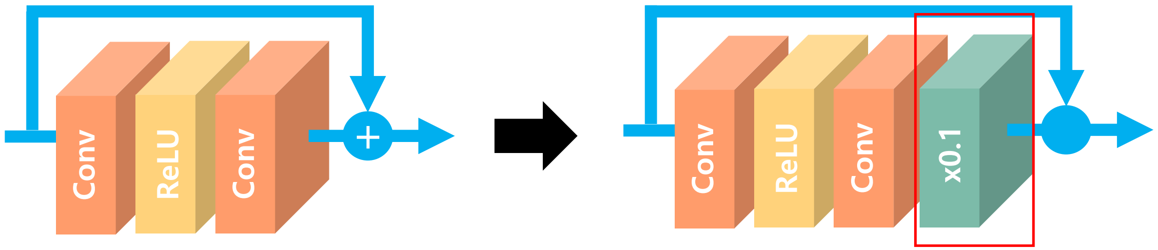
  → build up larger model

EDSR



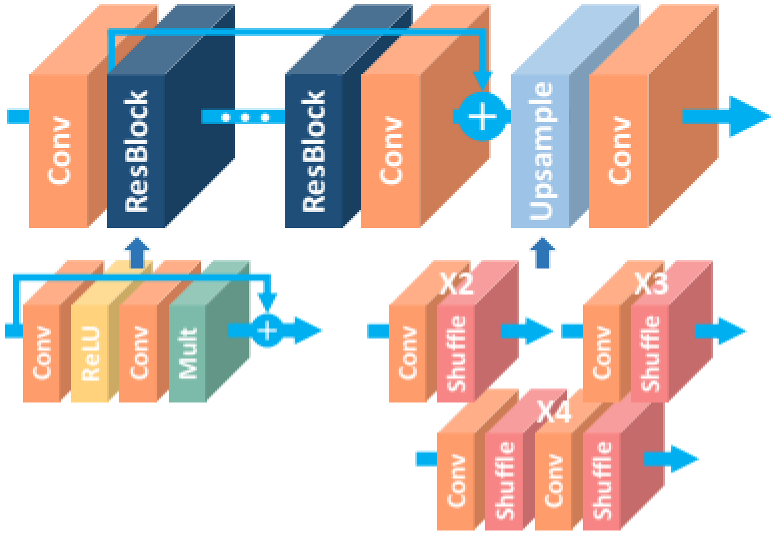
Single-scale SR network (EDSR)

❖ Adopting the residual scaling in residual block for larger model



- Motivation:
  - when increasing number of feature→ increase performance
    → make the training procedure numerically unstable
- Effect:
  - constant scaling layers with factor 0.1 when F=256
    → stabilize the training procedure

EDSR



Single-scale SR network (EDSR)

❖ Using L1 loss function instead of L2

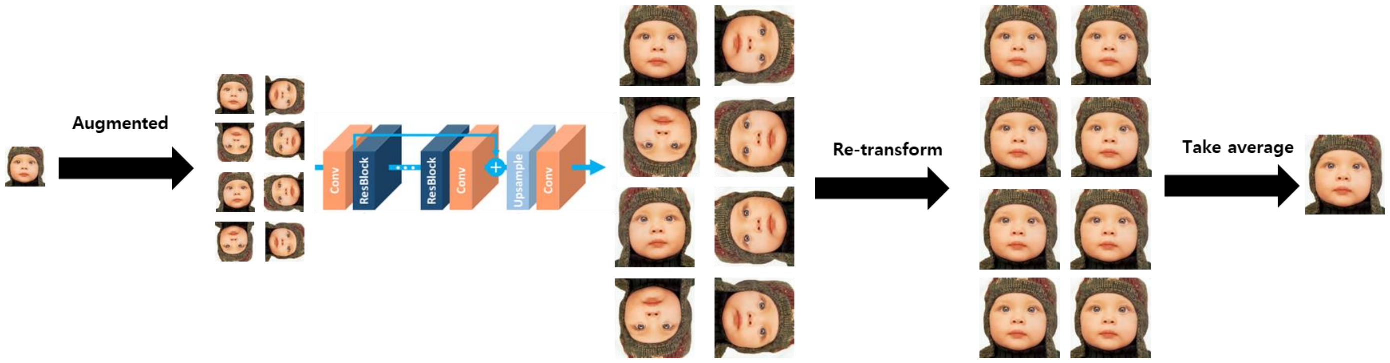
Performance comparison between L1 and L2 loss function

Scale	SRResNet (L2 loss)	SRResNet (L1 loss)
$\times 2$	34.40 / 0.9662	34.44 / 0.9665
$\times 3$	30.82 / 0.9288	30.85 / 0.9292
$\times 4$	28.92 / 0.8960	28.92 / 0.8961

- Motivation:
  - If |error| <1, the model will see much larger error in L1 norm than L 2 (|e| > e^2) → the model is more sensitive.
  - L1 loss provides better convergence than L2 in experiment

EDSR

❖ Self-ensemble

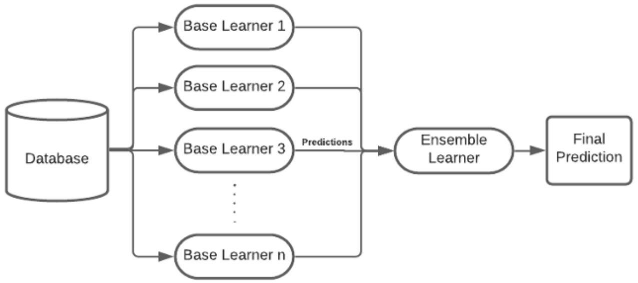


Self-ensemble for SR

Performance when applying self-ensemble.

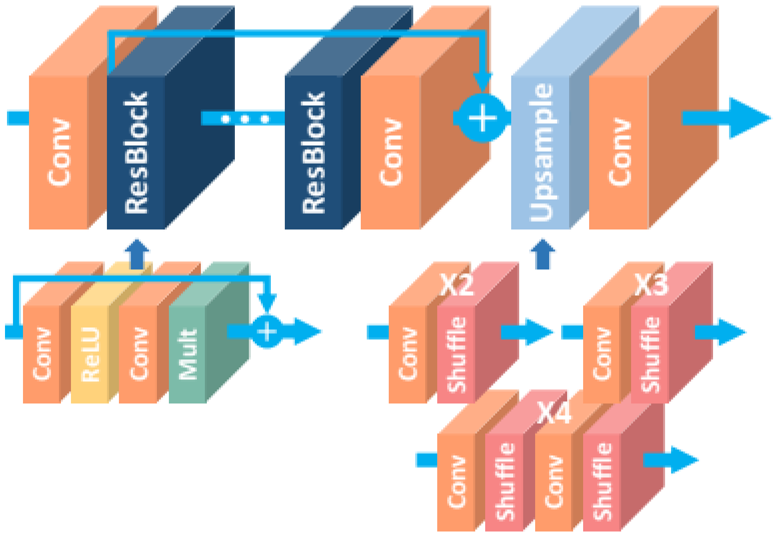
Dataset	Scale	EDSR (Ours)	**EDSR+ (Ours)**
Set5	×2	38.11 / 0.9601	38.20 / 0.9606
	×3	34.65 / 0.9282	34.76 / 0.9290
	×4	32.46 / 0.8968	32.62 / 0.8984
Set14	×2	33.92 / 0.9195	34.02 / 0.9204
	×3	30.52 / 0.8462	30.66 / 0.8481
	×4	28.80 / 0.7876	28.94 / 0.7901
B100	×2	32.32 / 0.9013	32.37 / 0.9018
	×3	29.25 / 0.8093	29.32 / 0.8104
	×4	27.71 / 0.7420	27.79 / 0.7437
Urban100	×2	32.93 / 0.9351	33.10 / 0.9363
	×3	28.80 / 0.8653	29.02 / 0.8685
	×4	26.64 / 0.8033	26.86 / 0.8080
DIV2K validation	×2	35.03 / 0.9695	35.12 / 0.9699
	×3	31.26 / 0.9340	31.39 / 0.9351
	×4	29.25 / 0.9017	29.38 / 0.9032

- Motivation:
  - keeps the total number of parameters same but increasing performance
  - Same performance gain compared to conventional model ensemble method that requires individually trained model



Ensemble Learning for Deep Learning Networks

EDSR

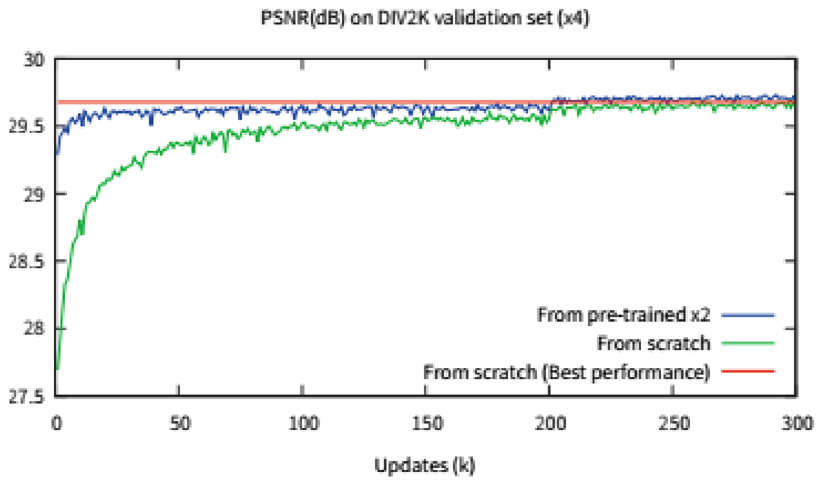


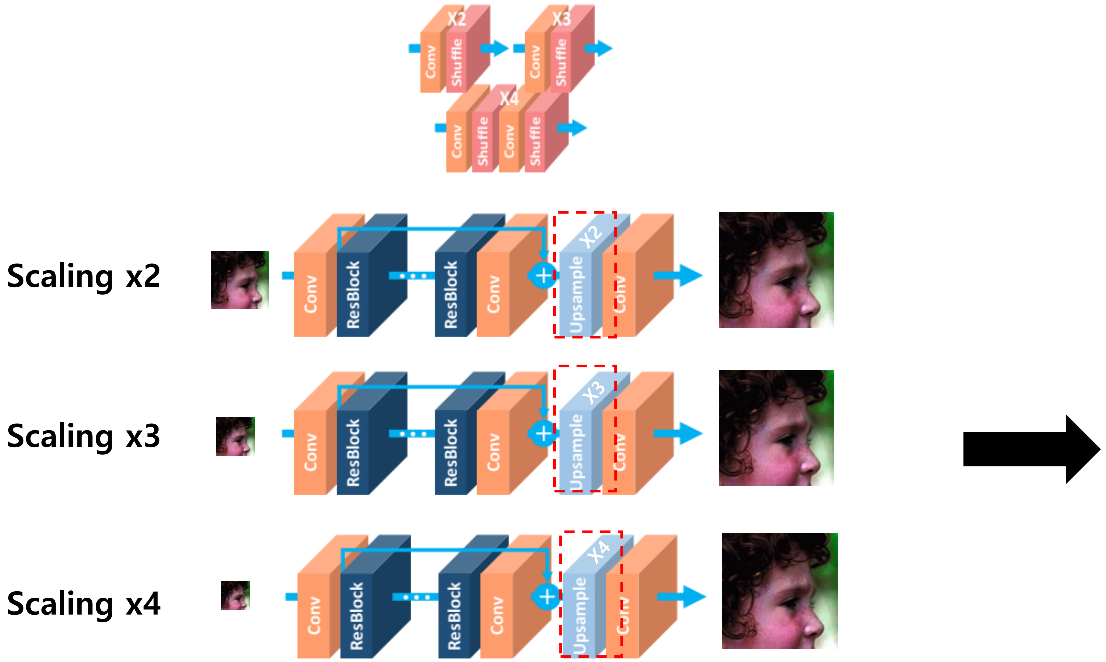
Single-scale SR network (EDSR)

- Structure: 32 residual blocks and 256 filters/layers
- Number of params: 43M
- Residual scaling 0.1
- Global and local skip connections
- Post-upscaling by pixel shuffle layer
- Remove BN
- L1 loss function
- Geometric self-ensemble (EDSR+)
- Single-scale

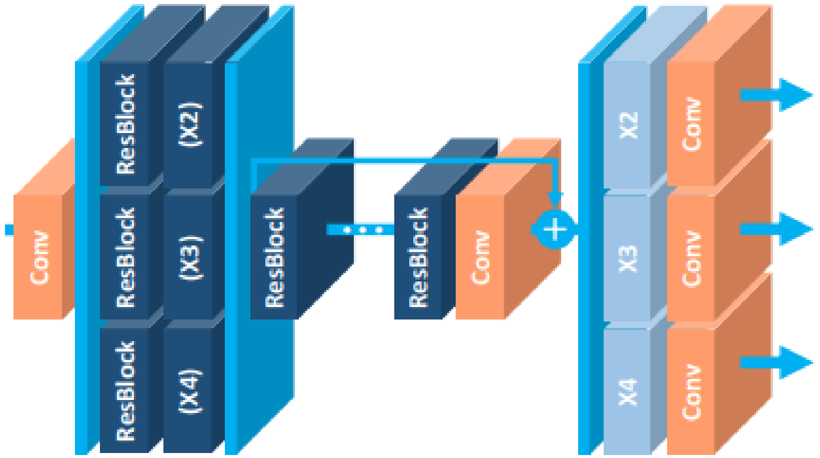
MDSR

- Motivation:
  - Most of existing methods use different structure corresponding to upscale factors →Training burden
  - SR at multiple scales is inter-related tasks





Single-scale SR network (EDSR)



Multi-scale SR network (MDSR)

MDSR

❖ Structure

- Pre-processing module

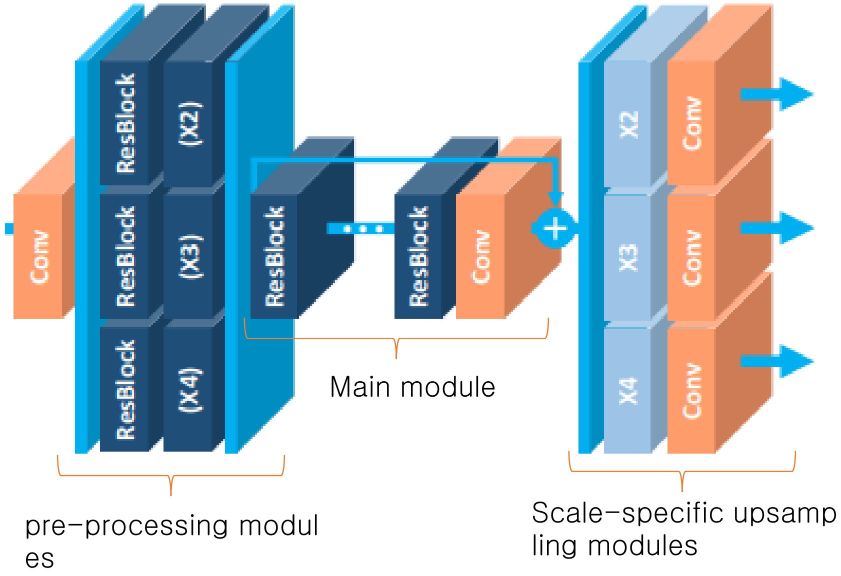
→ reduce the variance from input images of different scales.

- Main module

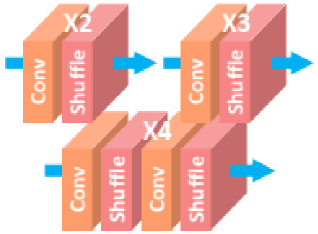
→ Sharing parameters

- Scale-specific upsampling modules

→ Generate output with specific scale



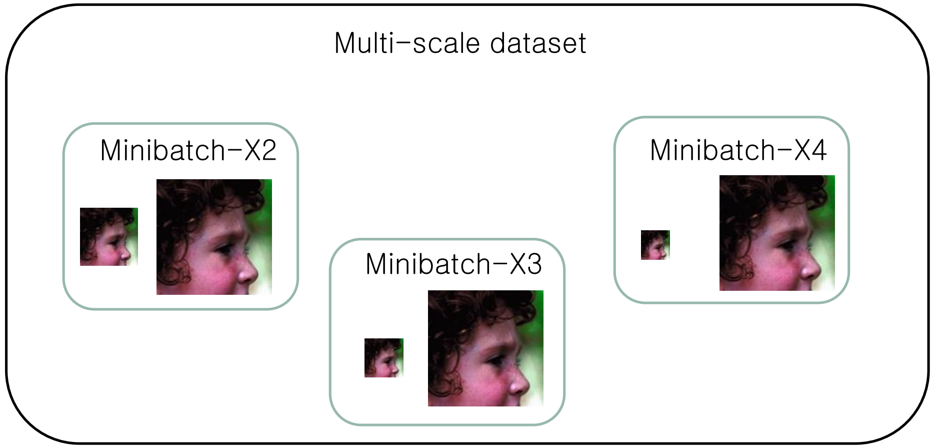
- Effect:
  - Reduce parameters totaling 4.5M (3x1.5M EDSR) to 3.2M (1 MDSR) in baseline structure
  - Exhibits comparable performance as the single-scale models

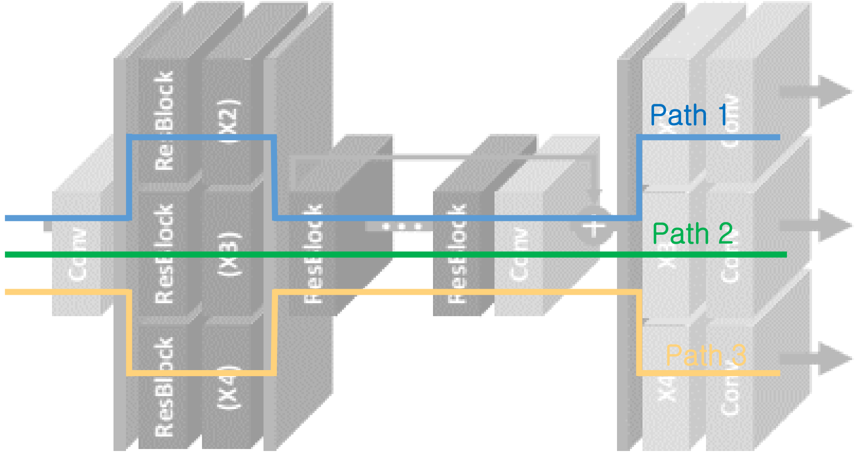


MDSR

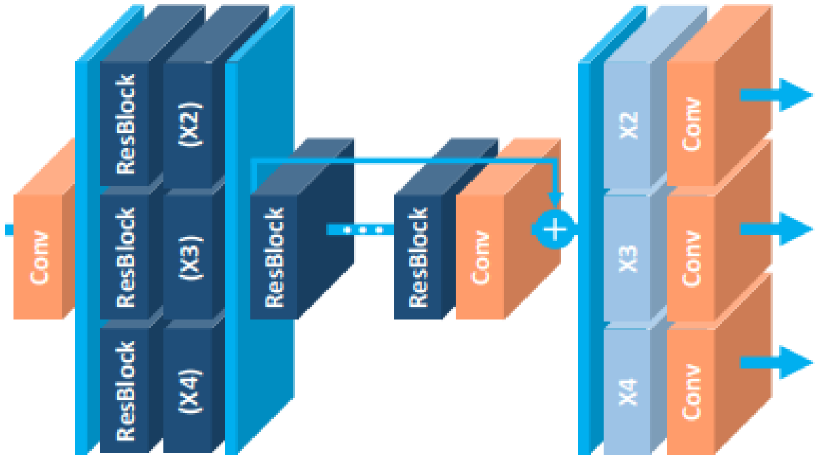
❖ Training and testing

- Training dataset:
  - multi-scale dataset including multi-scaling LR-HR pair
- Training
  - In each iteration, randomly selecting minibatch with specific scale among x2,x3,x4
  - Only modules corresponding to the scaling factor of taken mini batch are enable and learned
- Testing
  - Use the path corresponding to scaling factor of testing set.





MCSR



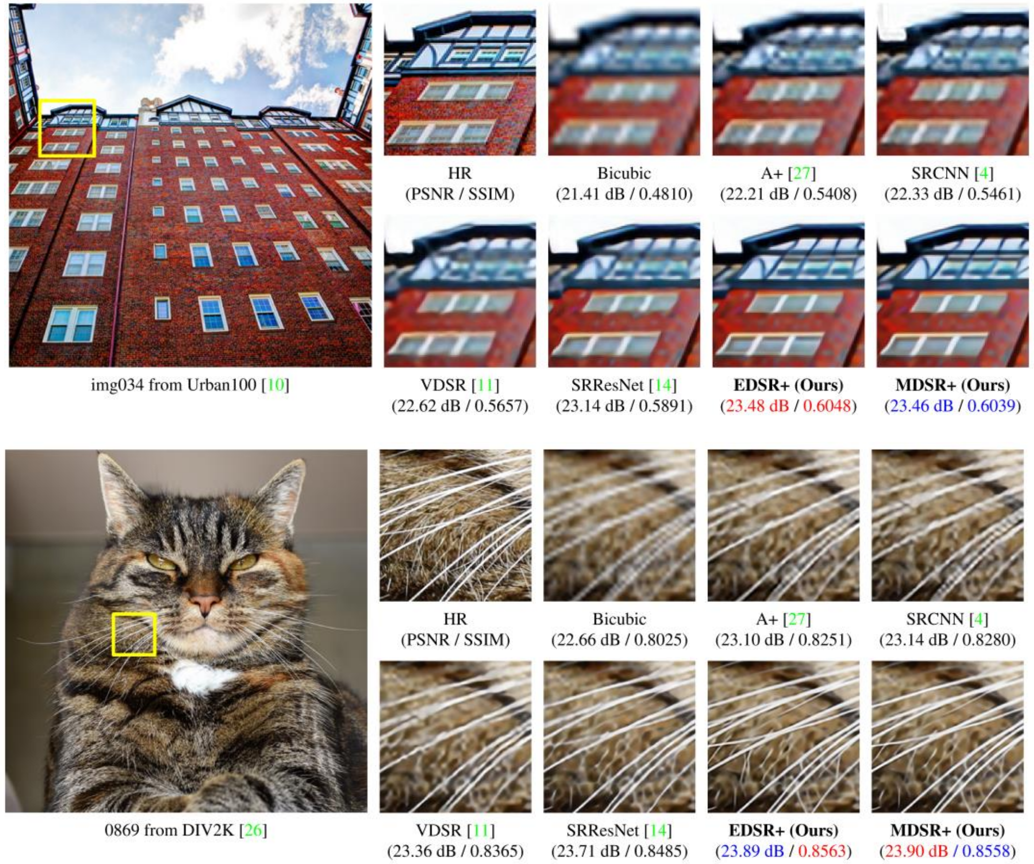
- Structure: 80 residual blocks and 64 filters/layers
- Number of params: 8M
- No residual scaling
- Global and local skip connections
- Post-upscaling by pixel shuffle layer
- Remove BN
- L1 loss function
- Geometric self-ensemble (MDSR+)
- Multi-scale

Experiments

- Training setting:
  - Dataset: DIV2K (800 images)
  - Data augmentation: random flips and rotations
  - Learning rate: 1e-4, halved at every 2e5 iterations
  - Optimizer: ADAM
  - Loss: L1

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

MCSR



Qualitative comparison of our models with other works on ×4 super-resoluti
on.

Conclusion

- Main contribution:
  - Remove BN → compact model
  - Using L1 loss function → enhance performance
  - Residual scaling techniques → stabilize large model training
  - Multi-scale SR network → save model size and training time