

2020.10.18

J. W. Soh, S. Cho and N. I. Cho, "Meta-Transfer Learning for Zero-Shot Super-Resolution," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 3513-3522, doi: 10.1109/CVPR42600.2020.00357.

### **Motivation:**

基于外部数据集的 SR 网络不能充分利用特定图像内部信息，并且只能用于特定条件下（如：Bicubic）的超分。为解决上述问题，ZSSR 提出利用图像内部信息来学习，但是 ZSSR 要求成千上万次梯度更新，导致很长的推理时间；同时仅依靠图像内部信息，存在训练样本少的问题。此外，元-学习能够解决人工智能无法使用少量样本去学习新任务的问题。

基于上述分析，本文在 ZSSR 的基础上，提出用于零次超分的元迁移学习 (MZSR)，它基于能够找到适合图像内部学习的通用初始参数。使用这种方法，能够利用外部图像和内部图像信息，在测试阶段仅需要很少的梯度更新次数即可产生可观的结果。

### **Contribution:**

提出一种基于元-迁移学习的训练策略，为零次无监督网络快速适应新任务学习了一个有效地初始权重参数；通过联合使用外部和内部样本，能够利用内部和外部数据各自的优势；提出的元迁移学习具有快速、灵活、轻量和无监督的特点，可应用于实际场景。

### **Introduction:**

网络结构分三个部分：大规模数据训练、元迁移学习、元测试。

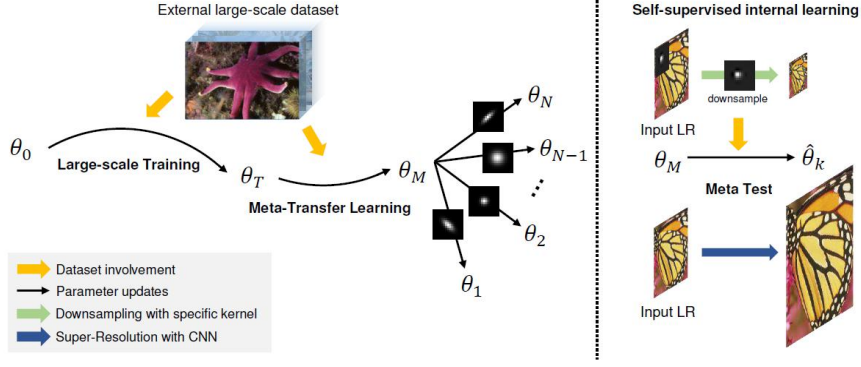


Figure 2: The overall scheme of our proposed MZSR. During meta-transfer learning, the external dataset is used, where internal learning is done during meta-test time. From random initial point  $\theta_0$ , large-scale dataset DIV2K [1] with “bicubic” degradation is exploited to obtain  $\theta_T$ . Then, meta-transfer learning learns a good representation  $\theta_M$  for super-resolution tasks with diverse blur kernel scenarios. The figure shows  $N$  tasks for simplicity. In the meta-test phase, self-supervision within a test image is exploited to train the model with corresponding blur kernel.

### 算法分析:

#### Algorithm 1: Meta-Transfer Learning

**Input:** High-resolution dataset  $\mathcal{D}_{HR}$  and blur kernel distribution  $p(k)$   
**Input:**  $\alpha, \beta$ : learning rates  
**Output:** Model parameter  $\theta_M$

- 1 Randomly initialize  $\theta$
- 2 Synthesize paired dataset  $\mathcal{D}$  by bicubically downsample  $\mathcal{D}_{HR}$
- 3 **while** not done **do**
- 4     Sample LR-HR batch from  $\mathcal{D}$
- 5     Evaluate  $\mathcal{L}^D$  by Eq. 2
- 6     Update  $\theta$  with respect to  $\mathcal{L}^D$
- 7 **end**
- 8 Generate task distribution  $p(\mathcal{T})$  with  $\mathcal{D}_{HR}$  and  $p(k)$
- 9 **while** not done **do**
- 10     Sample task batch  $\mathcal{T}_i \sim p(\mathcal{T})$
- 11     **for all**  $\mathcal{T}_i$  **do**
- 12         Evaluate training loss ( $\mathcal{D}_{tr}$ ):  $\mathcal{L}_{\mathcal{T}_i}^{tr}(\theta)$
- 13         Compute adapted parameters with gradient descent:  $\theta_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}^{tr}(\theta)$
- 14     **end**
- 15     Update  $\theta$  with respect to average test loss ( $\mathcal{D}_{te}$ ):  
 $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}^{te}(\theta_i)$
- 16     **end**
- 17 **end**

$$\mathcal{L}^D(\theta) = \mathbb{E}_{\mathcal{D} \sim (\mathbf{I}_{HR}, \mathbf{I}_{LR}^{bic})} [\|\mathbf{I}_{HR} - f_{\theta}(\mathbf{I}_{LR}^{bic})\|_1], \quad (2)$$

#### Algorithm 2: Meta-Test

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

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

### Result:

Dataset	Supervised			Unsupervised		
	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

Table 1: The average PSNR/SSIM results on “bicubic” downsampling scenario with  $\times 2$  on benchmarks. The numbers in parenthesis in our methods stand for the number of gradient updates.

Table 1 结果显示 MZSR 的性能优于 ZSSR，且 MZSR 仅需要很少的梯度更新。

Kernel	Dataset	Supervised			Unsupervised		
		Bicubic	RCAN [45]	IKC [11]	ZSSR [34]	MZSR (1)	MZSR (10)
$g_{0.2}^d$	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
$g_{2.0}^d$	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
$g_{ani}^d$	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
$g_{1.3}^b$	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

Table 2: The average PSNR/SSIM results on various kernels with  $\times 2$  on benchmarks. The numbers in parenthesis in our methods stand for the number of gradient updates. The best results are highlighted in red and the second best are in blue.

Table 2 结果显示对于使用非 Bicubic 模糊核的 LR 图像进行超分时，与相比监督方式，非监督方式性能更好，泛化能力更强。

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

Table 4: Comparisons of the number of parameters and time complexity for super-resolution of  $256 \times 256$  LR image with scaling factor  $\times 2$ .

Table 4 结果显示：与 SR 效果比较好的监督方法相比，非监督方法所需参数量很少，减少了内存占用；同时 MZSR 所需 SR 时间远小于 ZSSR，保证了 SR 的实时性。

### **Gain:**

本文通过联合内部外部图像信息提高了 ZSSR 的超分精度，使用预训练参数使得 ZSSR 能够快速收敛，减少了测试时间。通过使用元学习思想以适应不同的模糊核，增加了泛化能力；使用迁移学习思想将外部预训练参数作为无监督 SR 的初始权重。

2020.10.11

## 关于 ZSSR 论文问题:

*A. Shocher, N. Cohen and M. Irani, "Zero-Shot Super-Resolution Using Deep Internal Learning," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 3118-3126, doi: 10.1109/CVPR.2018.00329.*

重新读了一下 ZSSR 论文及其代码，发现 ZSSR 并没有使用预训练的参数，它将测试图像 LR 送入网络，采用从头训练的方式，其 SR 效果和监督方法接近。

## 阅读论文:

*G. Seif and D. Androutsos, "Large Receptive Field Networks for High-Scale Image Super-Resolution," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, 2018, pp. 876-87609, doi: 10.1109/CVPRW.2018.00120.*

## Motivation:

SR是一个不适定问题，存在一张LR图像对应多张HR图像问题。LR中包含HR中较少的高频细节，这在缩放因子较大时尤为明显。论文《K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.》（提出VGG网络）提出增加深度神经网络的感受野可以捕获更多的空间内容，进而提高网络性能。根据这一思路，很多SR工作通过使用更多的网络层数和递归来增加网络的深度，进而形成大的感受野，学习更复杂的image-to-image的映射。上面的做法存在问题：产生大量的网络参数、牺牲了网络速度、训练困难（需仔细设计学习率和梯度裁剪）。

针对上述问题，本文试图在不增加网络层数和参数数量的情况下，增大SR网络的感受野，进而提高SR网络性能。

## Contribution:

本文提出了两种增大感受野的方法：一维可分离滤波器(one-dimensional separable filters)；空洞卷积(atrous convolutions)。一维可分离滤波器通过使用更大的一维卷积核来增大感受野；空洞卷积使用卷积滤波器权重之间的空间来增大

感受野。两种方法相比使用标准卷积，增大了感受野，在不增加网络深度和参数数量情况下，提升了 SR 网络性能。

Algorithm	Scale	Depth	Params.	Set5	Set14	BSDS100	Urban100
Bicubic	$\times 4$	-	-	28.43 / 0.811	26.01 / 0.704	25.97 / 0.670	23.14 / 0.657
SCRNN [7]		3	57k	30.50 / 0.863	27.49 / 0.750	26.90 / 0.710	24.52 / 0.722
VDSR [13]		20	665k	31.35 / 0.883	28.01 / 0.767	27.29 / 0.725	25.18 / 0.752
DRCN [14]		20	1775k	31.54 / 0.884	28.03 / 0.768	27.24 / 0.725	25.14 / 0.752
LapSRN [16]		24	812k	31.54 / 0.885	28.19 / 0.772	27.32 / 0.727	25.21 / 0.756
DRRN [24]		52	297k	31.68 / 0.888	28.21 / 0.772	27.38 / 0.728	25.44 / 0.764
Baseline		26	889k	31.68 / 0.888	28.29 / 0.775	27.36 / 0.729	25.45 / 0.764
LRFNet-S		26	889k	31.91 / 0.890	28.44 / 0.778	27.47 / 0.733	25.70 / 0.773
LRFNet-A	$\times 8$	26	889k	31.82 / 0.889	28.38 / 0.777	27.39 / 0.730	25.61 / 0.769
Bicubic		-	-	24.40 / 0.658	23.10 / 0.566	23.67 / 0.548	20.74 / 0.516
SCRNN [7]		3	57k	25.33 / 0.690	23.76 / 0.591	24.13 / 0.566	21.29 / 0.544
VDSR [13]		20	665k	25.93 / 0.724	24.26 / 0.614	24.49 / 0.583	21.70 / 0.571
DRCN [14]		20	1775k	25.93 / 0.723	24.25 / 0.614	24.49 / 0.582	21.71 / 0.571
LapSRN [16]		36	812k	26.15 / 0.738	24.35 / 0.620	24.54 / 0.586	21.81 / 0.581
DRRN [24]		52	297k	26.18 / 0.738	24.42 / 0.622	24.59 / 0.587	21.88 / 0.583
Baseline		26	889k	26.46 / 0.753	24.46 / 0.626	24.63 / 0.589	21.98 / 0.590
LRFNet-S		26	889k	26.66 / 0.763	24.58 / 0.629	24.68 / 0.589	22.06 / 0.594
LRFNet-A		26	889k	26.77 / 0.765	24.68 / 0.631	24.73 / 0.590	22.09 / 0.594

Table 7: Test results on benchmark datasets. Red indicates the best performance and blue indicates the second best. Here, LRFNet-S uses a kernel size of 9 and LRFNet-A uses the 1-4-8 scheme.

# Introduction:

基本的网络结构:

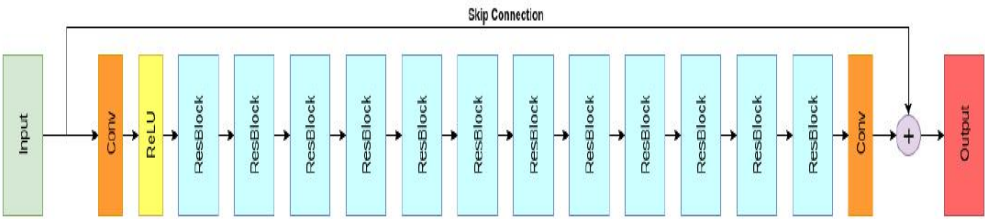


Figure 1: Our baseline model containing 12 residual blocks. Each residual block is composed of a conv-relu-conv structure and an additive skip connection.

基本网络中 ResBlocks 的结构:



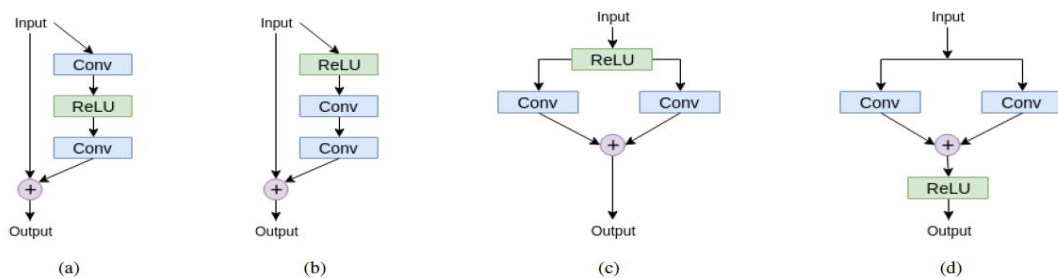


Figure 3: Exploring the network design space of local ResBlocks with separable filters. (a) Baseline residual block use in EDSR [19]. For the baseline model, we use this scheme with  $k \times k$  convolutions. For our model with one-dimensional separable filters, the first convolution is  $1 \times k$  and the second is  $k \times 1$ . (b) Moving the ReLU to before the separable filters. We test this to see if allowing the feature maps to pass through the vertical and horizontal convolutions uninterrupted (i.e without the activation inbetween) improves performance. (c) and (d) Adding the feature maps processed by the  $1 \times k$  and the  $k \times 1$ . We test this to see if adding the feature maps from the  $1 \times k$  and  $k \times 1$  convolutions is more effective due independent processing in the vertical and horizontal directions. We test placing the activation before (c) and after (d) the convolutions.

使用标准卷积核时，ResBlock 使用(a)结构是最优的，EDSR 中使用了这种结构；使用一维可分离卷积时，(a)结构仍然是最优的。下图中 A、B、C、D 依次表示上述四种结构。

Scheme	Scale	PSNR / SSIM
A	$\times 4$	<b>28.32 / 0.804</b>
	$\times 8$	<b>24.76 / 0.670</b>
B	$\times 4$	28.28 / 0.803
	$\times 8$	24.74 / 0.670
C	$\times 4$	28.27 / 0.802
	$\times 8$	24.73 / 0.668
D	$\times 4$	28.27 / 0.803
	$\times 8$	24.73 / 0.668

Table 1: ResBlock design for 1-D separable kernels. Tests are run on the NTIRE validation set.

三种卷积（标准卷积核、一维可分离卷积核、空洞卷积核）：

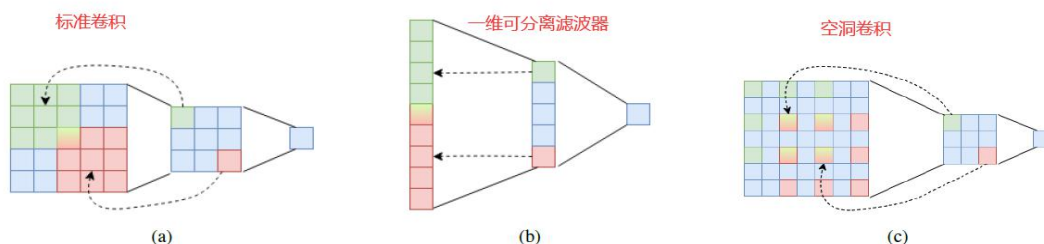


Figure 2: Network receptive field from two successive convolutions. (a) With two  $3 \times 3$  convolutions we get an overall receptive field of 5 in both the vertical and horizontal directions. (b) Here we use two  $5 \times 1$  convolutions and get an overall receptive field of 9 in the vertical direction, much larger in the vertical than using a square  $3 \times 3$  but with less parameters (10 total vs 18 total in the  $3 \times 3$  case). (c) Two successive  $3 \times 3$  convolutions but the second convolution has a dilation rate of 2. This expands the overall receptive field from 5 in both directions (in (a)) to 7 in both directions without increasing the number of parameters.

一维可分离卷积(LRFNet-S)：

标准卷积和一维可分离卷积结果比较：在 SR 网络中使用的多是标准卷积核（如：3\*3 或 5\*5），由 Table 2 中的 Baseline 知，标准卷积核的 size 为 7\*7 时性

能最优，随着 size 增大网络性能先变好再变差；LRFNet-S 使用一维可分离卷积核（如：1\*3,3\*1 或 1\*5,5\*1），随着卷积核 size 增大，网络性能增加。由 Table 3 知使用一维可分离卷积核相较标准卷积核，在很大程度上减小了网络的参数量。

Kernel Size	Scale	Baseline PSNR / SSIM	LRFNet-S PSNR / SSIM
3	× 4	28.26 / 0.801	28.07 / 0.795
	× 8	24.69 / 0.668	24.54 / 0.662
5	× 4	28.29 / 0.803	28.23 / 0.800
	× 8	24.74 / 0.670	24.69 / 0.668
7	× 4	<b>28.31 / 0.803</b>	28.34 / 0.804
	× 8	<b>24.75 / 0.670</b>	24.76 / 0.671
9	× 4	28.29 / 0.803	28.42 / 0.806
	× 8	24.72 / 0.670	24.77 / 0.671
11	× 4	28.25 / 0.800	<b>28.45 / 0.808</b>
	× 8	24.69 / 0.668	<b>24.79 / 0.672</b>

Table 2: Comparing the performance of using different kernel size with 1-D separable kernels and the baseline. LRFNet-S indicates the model with 1-D separable kernels. Tests are run on the NTIRE validation set.

Kernel Size	Baseline Params / Time	LRFNet-S Params / Time
3	889k / 0.989	299k / 0.854
5	2,462k / 2.14	496k / 1.00
7	4,821k / 3.68	693k / 1.08
9	7,967k / 6.39	889k / 1.21
11	11,899k / 8.38	1,086k / 1.35

Table 3: Comparing the parameter count and inference time of using different kernel size with 1-D separable convolutions and the baseline. LRFNet-S indicates the model with 1-D separable kernels. Time is in seconds. Tests are run on the NTIRE validation set.

空洞卷积(LRFNet-A):

Table 4 比较了标准卷积和不同空洞率的空洞卷积。空洞卷积实际卷积核大小:  $K=k+(k-1)(r-1)$ , k 为原始卷积核大小, r 为空洞卷积参数空洞率。表中 Baseline 表示使用标准卷积, 1-2 表示基本网络的 12 个 ResBlocks 中前 6 个空洞率为 1(空洞率为 1 时即标准卷积), 后六个空洞率为 2; 1-2-3 表示 ResBlocks 的前 4 个空洞率用 1, 中间 4 个用 2, 后 4 个用 3。由结果知, 在 scale 较大时, 使用空洞率较大的卷积 SR 效果更好。

Scheme	Scale	PSNR / SSIM	Time
Baseline	× 4	28.26 / 0.801	0.989
	× 8	24.69 / 0.668	
1-2	× 4	28.34 / 0.804	1.27
	× 8	24.74 / 0.670	
1-2-3	× 4	28.36 / 0.806	1.37
	× 8	24.80 / 0.672	
1-3-5	× 4	<b>28.38 / 0.806</b>	1.38
	× 8	24.81 / 0.672	
1-4-8	× 4	28.36 / 0.805	1.45
	× 8	<b>24.84 / 0.673</b>	

Table 4: Comparing the performance of using different dilation schemes with Atrous convolutions all using 3x3 kernels. Time is in seconds. Tests are run on the NTIRE validation set.

联合使用一维可分离卷积和空洞卷积(LRFNET-SA):

由 Table 5 知, 将两者联合使用结果并不会更好, 相比只是用一维可分离卷积, 联合使用效果会变差。

Kernel Size	Scale	LRFNet-SA PSNR / SSIM	LRFNet-S PSNR / SSIM
3	$\times 4$	28.04 / 0.794	28.07 / 0.795
	$\times 8$	24.50 / 0.660	24.54 / 0.662
5	$\times 4$	<b>28.16 / 0.798</b>	28.23 / 0.800
	$\times 8$	<b>24.62 / 0.665</b>	24.69 / 0.668
7	$\times 4$	28.14 / 0.797	28.34 / 0.804
	$\times 8$	24.60 / 0.664	24.76 / 0.671
9	$\times 4$	28.10 / 0.796	28.42 / 0.806
	$\times 8$	24.58 / 0.663	24.77 / 0.671
11	$\times 4$	28.03 / 0.793	<b>28.45 / 0.808</b>
	$\times 8$	24.54 / 0.661	<b>24.79 / 0.672</b>

Table 5: Comparing the performance of using different 1-D separable kernel sizes with dilation. All models use the same dilation rate scheme of 1-4-8. Time is in seconds. Tests are run on the NTIRE validation set.

### 比较一维可分离卷积和空洞卷积：

空洞卷积在 scale 较大时，表现较好；一维可分离卷积在 scale 较小时，表现较好。

Algorithm	Scale	DIV2K Val. PSNR / SSIM
Baseline	$\times 4$	28.63 / 0.812
	$\times 8$	25.03 / 0.679
LRFNet-S	$\times 4$	28.76 / 0.816
	$\times 8$	25.14 / 0.682
LRFNet-A	$\times 4$	28.68 / 0.814
	$\times 8$	25.24 / 0.684

Table 6: Comparing the performance of our fully trained models on both  $\times 4$  and  $\times 8$  scales for the NTIRE validation set. LRFNet-S uses 1-D kernels and LRFNet-A uses atrous convolutions

### Gain:

目前研究的超分方法用的是标准卷积核，本文对此进行了研究，使用 kernel size 太小（3\*3）或者太大（11\*11）的卷积核都不利于 SR，可以尝试修改 SR 网络中的 kernel size 来优化。对于 scale 较大时，空洞卷积表现比较好，将来也可以用来尝试优化网络。



2020.10.5

项目研究进展:

### 1. 遇到的问题:

用 10 组  $256*64*50$  大小的不同深度的信道数据作为训练集, 在较大放大倍数下, SR 性能较差。推测可能是以下问题导致:

- ① 不同组信道数据是在不同深度下得到的, 因此不同组间数据分布量级差距较大
- ② 同一组信道数据的 50 个通道之间数据分布范围差距较大 (可由直方图看出)

### 2. 解决办法:

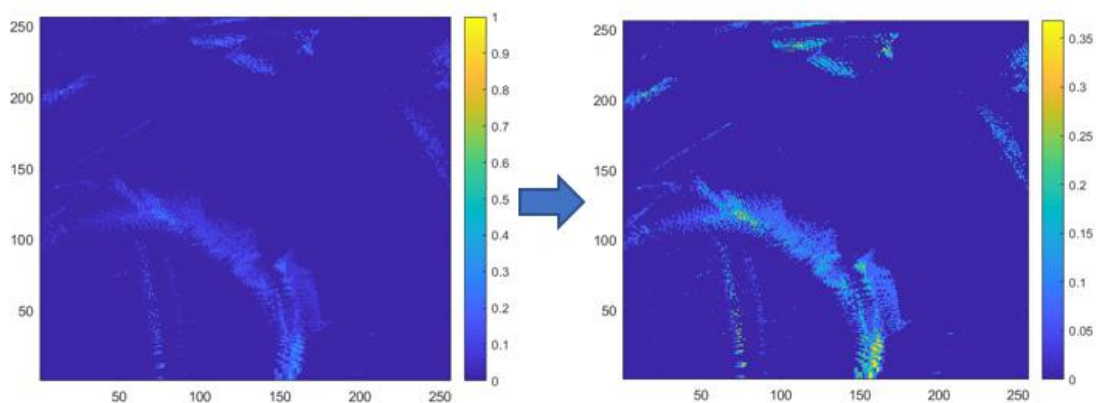
针对第一个问题: 初步的想法是既然不同组之间差距较大, 可以只用独立的一组信道数据作为训练集, 然后针对不同信道数据独立训练各种网络, 这样就不需要考虑组间数据量级差距大的问题。这样做有一个问题就是: 比如别人给你一组  $64*64*150$  的数据, 你想将他 SR 到  $256*256*150$  大小, 需要事先确定该用哪一个 SR 网络。还有就是要用多少组信道数据, 训练多少个独立的网络。

对于第二个问题: 采用数据扩展的思想。将同一组数据的不同通道, 根据每个通道的直方图分布, 对其扩展。具体的操作在实验部分讲解。

**为了验证上述想法做了以下实验:**

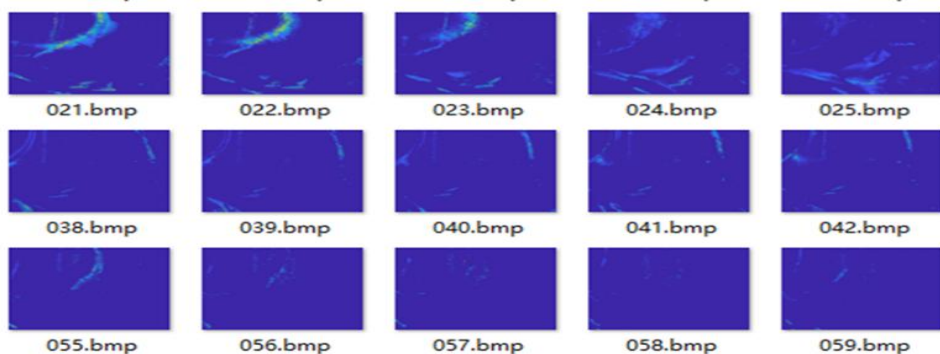
生成了一组  $256*256*150$  的信道数据, 训练一个独立的网络。数据包含 150 个通道, 这 150 个通道也存在数据分布范围差别大的问题, 针对这一问题, 利用 matlab 中 `imagesc` 函数, 根据每一个通道 (大小为  $256*256$ ) 的 min 和 max 值, 将其映射到相同的色彩空间, 这样可以解决同一组信道数据分布范围差距大问题。

如下图所示, 用 `imagesc` 函数根据矩阵分布将其映射到固定的色彩空间。图像的横纵轴表示  $256*256$  大小的通道矩阵, 图中的颜色与比例尺对应。这一变换相当于对直方图做了扩展处理。



数据分布扩展

将一组  $256 \times 256 \times 150$  的信道数据视为 150 张  $256 \times 256$  的矩阵，利用 `imagesc` 函数将矩阵转变为图像同时对直方图扩展，得到如下数据集：



150 张数据集（随机选取一部分）

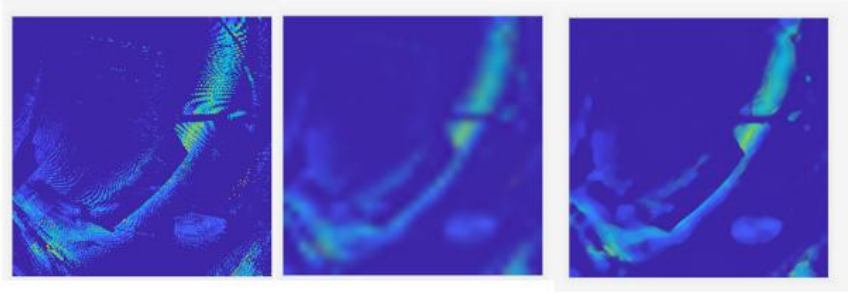
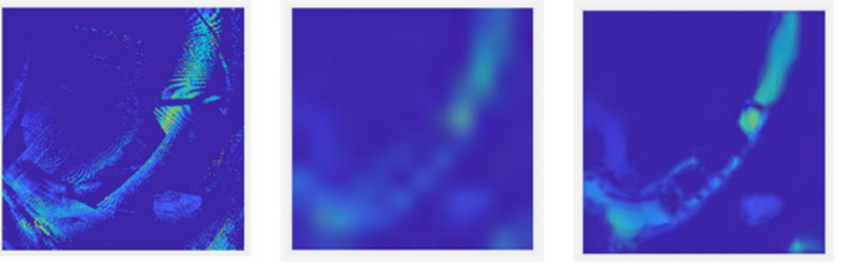
用上述数据集训练了一个独立的网络，并用数据集中的部分图像测试网络 SR 效果，但是 SR 效果不理想。

### 分析：

在上述构造的数据集中已经不存在上文中的两个问题，理论上讲用一组信道数据训练网络并用训练集中的图像测试网络应该能取得较好的效果。

SR 效果一般的原因：数据集的问题，有些图像几乎不包含高频信息，网络从 LR 到 HR 的映射中很难学到知识；常规图像超分中数据集包含各种各样的图像，包含更多的结构信息，测试图像能在其中找到接近的结构，泛化能力更强。

由于无监督 ZSSR 网络在性能上和监督 RCAN 网络相差不大，ZSSR 只需要使用一张 LR 图像即可做 SR，所以用 ZSSR 测试了一下对信道图像做 SR 的效果：

Scale	HR	Bicubic	SR
X8			
X20			

分析：

ZSSR 在较大 Scale 时，SR 效果优于 Bicubic，推测：如果我们能构造出合适的训练集，用监督 SR 方法在大 scale 下应该也能获得较好的 SR 效果。

### 3. 总结

到现在为止，都是基于思路一来做的，即：不考虑同一信道相邻采样点的相关性，将  $256 \times 256 \times 150$  大小的数据当做 150 张  $256 \times 256$  大小的单通道图像作为训练集，训练 SR 网络。将 LR 数据依次送入网络做 SR，将 SR 后的图像整合，实现 SR 信道数据的目的。

根据上述思路做的工作：

使用不同深度的多组数据训练一个能够适应各种应用场景的超分网络，类似常规图像超分，训练的网络可以适用于各种图像。基于这一思想：利用 10 组  $256 \times 64 \times 50$  的数据进行了测试，SRCNN 和 Lapsrn 网络在 Scale（缩放因子）较小时 (2, 4, 8)，超分网络有增益，RCAN 和 EDSR 网络在  $\text{scale}=4$  时增益就不明显了。针对大 scale 超分增益差的问题，研究了 patchsize 对 SR 的影响以及现有的信道数据是否适合 SR。

**Patchsize:** patchsize 的选取对 SR 是有影响的，因为当 patchsize 很小的时候，其对应的 LR ( $\text{patchsize}/\text{scale}$ ) 将会非常小，尤其是当 scale 很大的时

候，这就会导致不定问题（多张 HR 图像对应一张 LR 图像），另外信道数据本身零值太多比较稀疏，更加剧了这一问题。

**信道数据：**由于一组信道数据的不同通道之间数据分布范围差别很大，这对网络的训练和后面的信道恢复是有影响的，对此可以使用直方图扩展的办法将每个通道的直方图进行扩展。同时发现不同组（不同深度）的信道数据的量级差异也很大，对此可以将每个信道视作一个独立的数据集，对每一组信道数据训练一个独立的 SR 网络。

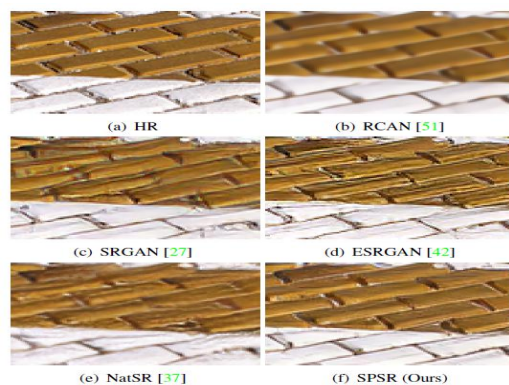
阅读论文：

C. Ma, Y. Rao, Y. Cheng, C. Chen, J. Lu and J. Zhou, "Structure-Preserving Super Resolution With Gradient Guidance," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 7766-7775.

## Motivation:

GAN 网络在 SR 中能够很好地恢复图像细节（其他 SR 方法存在模糊现象），但在保持图像结构方面表现不好，常常出现结构失真。本文在 GAN 的基础上进行改进。梯度图是反映图像结构信息的理想工具，基于图像梯度图的图像恢复方法能够较好地保持图像结构信息。因此本文在 GAN 上增加了一个图像梯度分支来实现更好的 SR。

几种 SR 图像的比较如图：相比 RCAN(b)，可以看出 GAN(c, d, e) 恢复细节较好，但存在结构失真，文章所提方法(f)吸收了 GAN 的优点同时克服了其缺点。



## Contribution:



1. 在 SR 分支的基础上增加了一个梯度分支，将 LR 图像的梯度图转换为 HR 图像的梯度图，恢复的梯度图整合到 SR 分支为 SR 提供结构先验。梯度分支的目标是估计梯度图像从 LR 到 HR 的转换。一张图像的梯度可以通过计算相邻像素的差值得到。

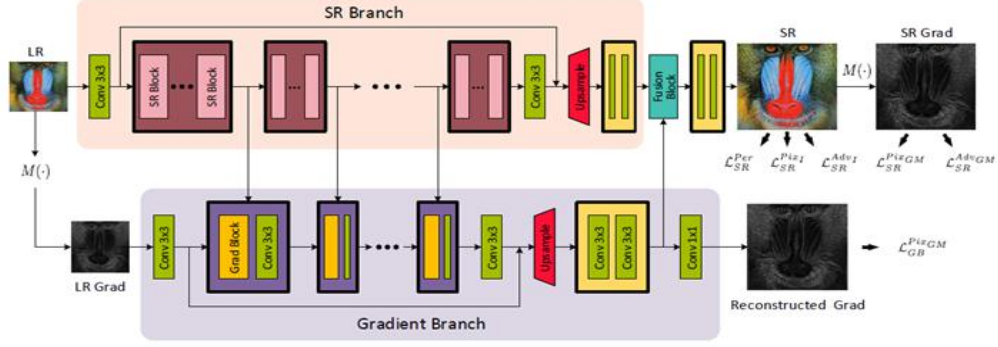


Figure 2. Overall framework of our SPSR method. Our architecture consists of two branches, the SR branch and the gradient branch. The gradient branch aims to super-resolve LR gradient maps to the HR counterparts. It incorporates multi-level representations from the SR branch to reduce parameters and outputs gradient information to guide the SR process by a fusion block in turn. The final SR outputs are optimized by not only conventional image-space losses, but also the proposed gradient-space objectives.

$$\begin{aligned}
 I_x(\mathbf{x}) &= I(x+1, y) - I(x-1, y), \\
 I_y(\mathbf{x}) &= I(x, y+1) - I(x, y-1), \\
 \nabla I(\mathbf{x}) &= (I_x(\mathbf{x}), I_y(\mathbf{x})), \\
 M(I) &= \|\nabla I\|_2,
 \end{aligned}$$

2. 提出了梯度损失，即最小化由 SR 图像和对应 HR 图像提取的梯度图的距离。通过联合梯度损失和图像 SR 损失，构成双重损失函数，生成器不仅能够学到细节信息，还能避免图像结构失真。

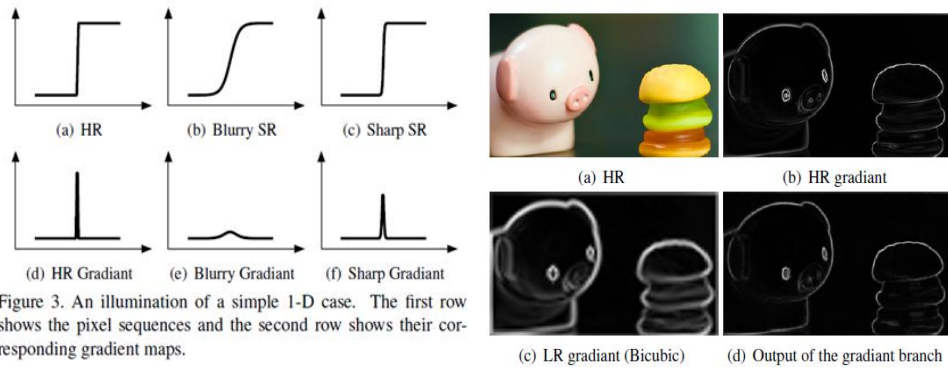


Figure 3. An illumination of a simple 1-D case. The first row shows the pixel sequences and the second row shows their corresponding gradient maps.

## Gain:

引入图像梯度图能够更好地保持图像的结构信息。信道数据结构信息本身不明显，无法获得好的图像梯度图，感觉很难用在信道数据上。

