Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network

Wenzhe Shi[1], Jose Caballero[1], Ferenc Huszár[1], Johannes Totz[1], Andrew P. Aitken[1],
Rob Bishop[1], Daniel Rueckert[1], Zehan Wang[1]
[1]Twitter
[1]{wshi,jcaballero,fhuszar,jtotz,aitken,rbishop,zehanw}@twitter.com

Presented: Hengtao Guo

09/26/2018

Background Information

1. Super-resolution(SR) problem: Low resolution(LR) to high resolution(HR)
2. A key assumption of SR: much of the high-frequency data is redundant and thus can be accurately reconstructed from low frequency components.
3. Methods are categorized into multi-image SR methods and **single image super-resolution (SISR)** methods.
4. Sparsity-based techniques: a dictionary of image atoms which can be learnt through training

Other works

1. Image resolution is increased in the middle of the network gradually.
2. Increase the resolution before or at the first layer of the network.
3. Drawbacks of these works:
   - Increase the computational complexity
   - Interpolation methods do not bring in useful information

This paper proposes

1. Learning the upscaling filters to compensate information
2. Increase the resolution only at the very end of the network
3. Super-resolve data from LR feature maps

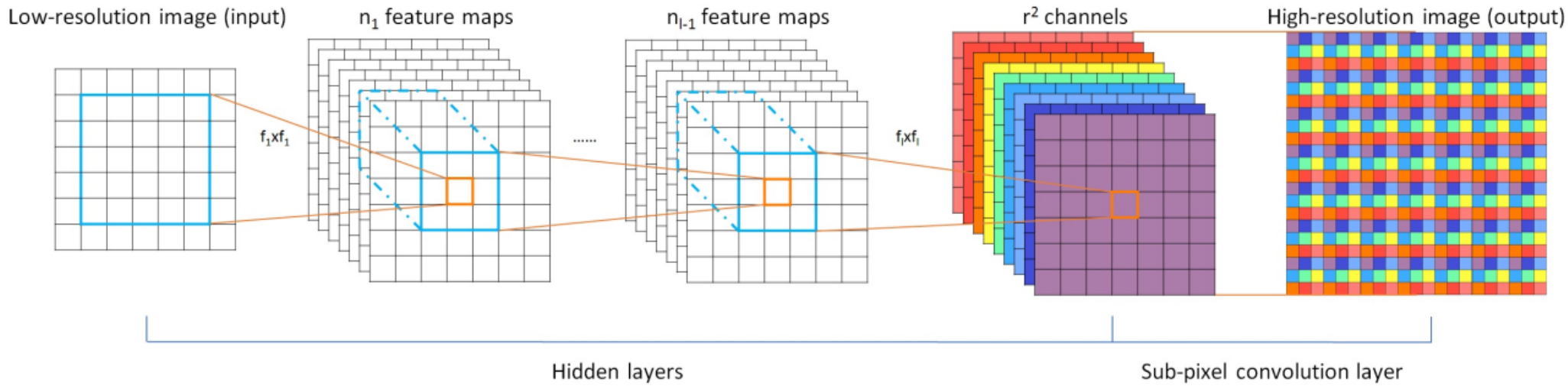


Figure 1. The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

Why we separate upscale filters?

1. Each LR image is directly fed to the network
2. Smaller image using smaller filter size, **lower the computational and memory complexity**. Strong even in real-time.
3. For a network with L layers, learn n_{L-1} upscaling filters for the n_{L-1} feature maps instead of one upscaling filter for the input image.
4. The network is capable of learning a better and **more complex LR to HR mapping** compared to a single fixed filter upscaling at the first layer.
5. Results in additional gains in the **reconstruction accuracy** of the model.

Method

1. Refer r as the scaling ratio
2. From $H \times W \times C$ to $rH \times rW \times C$
3. **First L-1 layers**: gather information

$$f^1(\mathbf{I}^{LR}; W_1, b_1) = \phi\left(W_1 * \mathbf{I}^{LR} + b_1\right), \quad (1)$$

$$f^l(\mathbf{I}^{LR}; W_{1:l}, b_{1:l}) = \phi\left(W_l * f^{l-1}\left(\mathbf{I}^{LR}\right) + b_l\right), \quad (2)$$

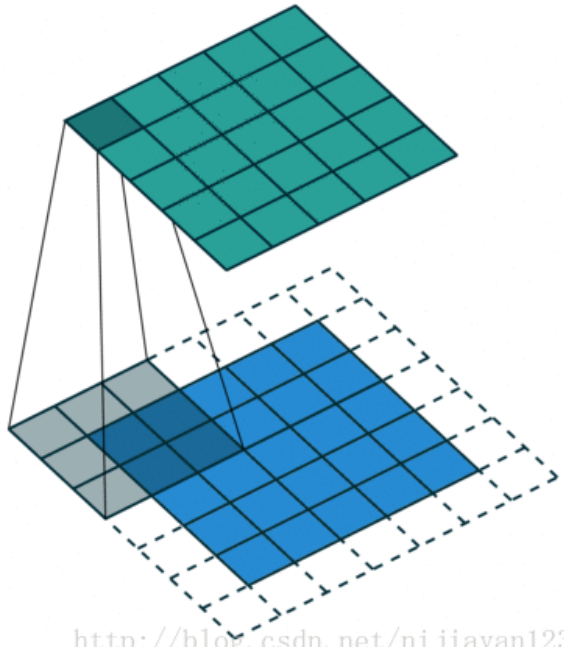
4. W_l and b_l are learnable network weights and biases, W_l size:

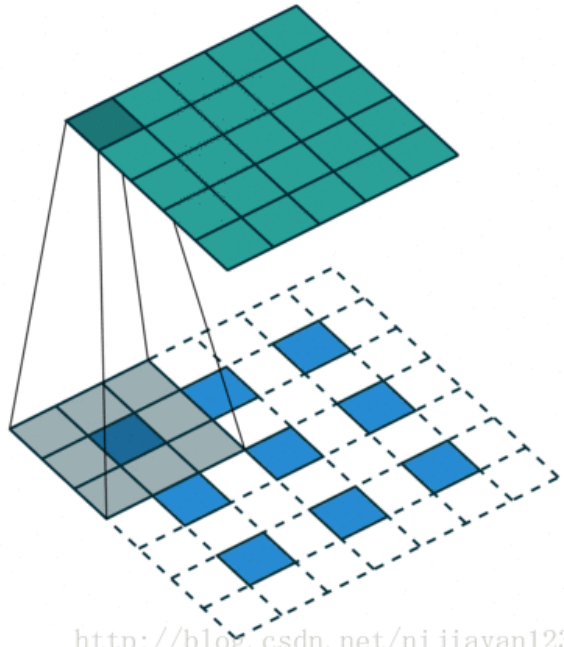
$$n_{l-1} \times n_l \times k_l \times k_l$$

5. **Last layer** f_L has to convert the LR feature maps to a HR image $\boldsymbol{I}^{SR}$

Last layer: upscale

- Common methods:
  1. Interpolation
  2. Perforate
  3. Un-pooling
  4. Deconvolution





Efficient sub-pixel convolution layer

1. A convolution with stride of $\frac{1}{r}$ with a filter W_s of size k_s
2. Last layer presentation: $\mathbf{I}^{SR} = f^L(\mathbf{I}^{LR}) = \mathcal{PS}\left(W_L * f^{L-1}(\mathbf{I}^{LR}) + b_L\right),$
3. Periodic shuffling operator that rearranges the elements of a $H \times W \times C \cdot r^2$ tensor to a tensor of shape $rH \times rW \times C$

$$\mathcal{PS}(T)_{x,y,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, C \cdot r \cdot mod(y,r) + C \cdot mod(x,r) + c}$$

4. Objective function using pixel-wise mean squared error (MSE)

$$\ell(W_{1:L}, b_{1:L}) = \frac{1}{r^2 HW} \sum_{x=1}^{rH} \sum_{x=1}^{rW} \left(\mathbf{I}^{HR}_{x,y} - f^L_{x,y}(\mathbf{I}^{LR})\right)^2$$

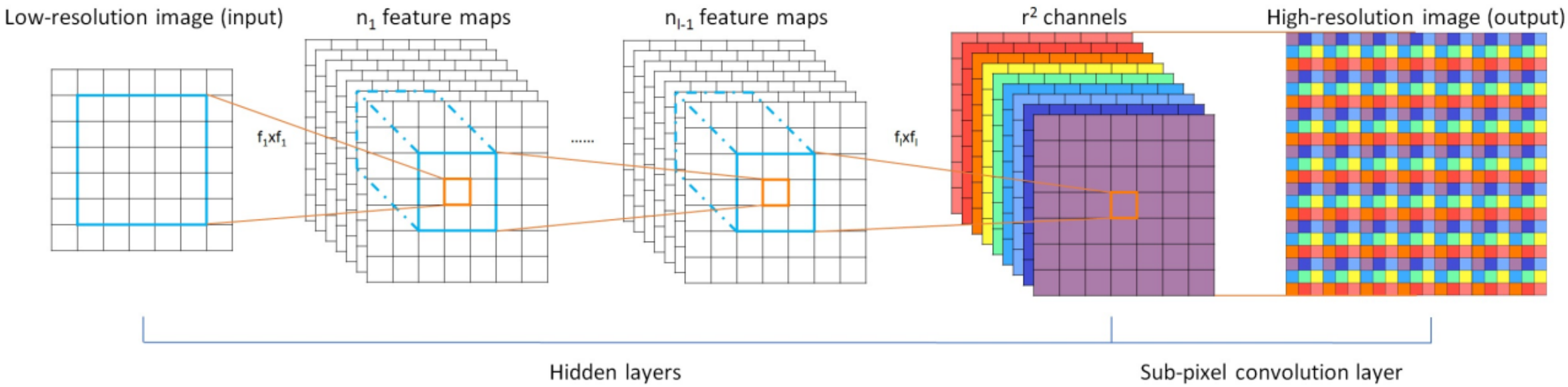


Figure 1. The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

Experiments

1. Total 3 layers, 2 for feature maps, 1 for super-resolution
2. Scaling of 3, inspired by SRCNN's architecture
3. Using tanh instead of RELU as the activation function
4. Stops when no improvements during100 epochs.
5. Initial learning rate is set to 0.01 and final learning rate is set to 0.0001 and updated gradually when the improvement of the cost function is smaller than a threshold μ.

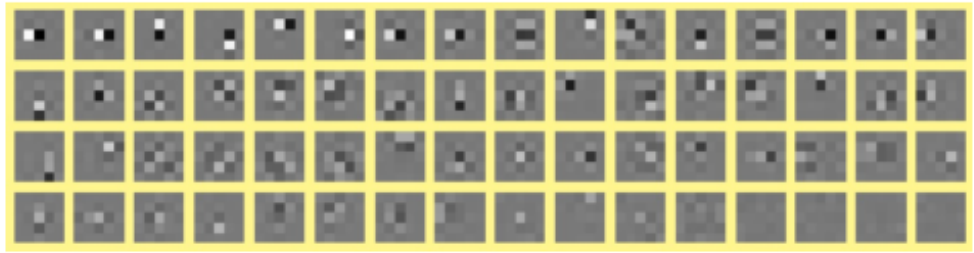


Figure 3. The first-layer filters trained on ImageNet with an upscaling factor of 3. The filters are sorted based on their variances.

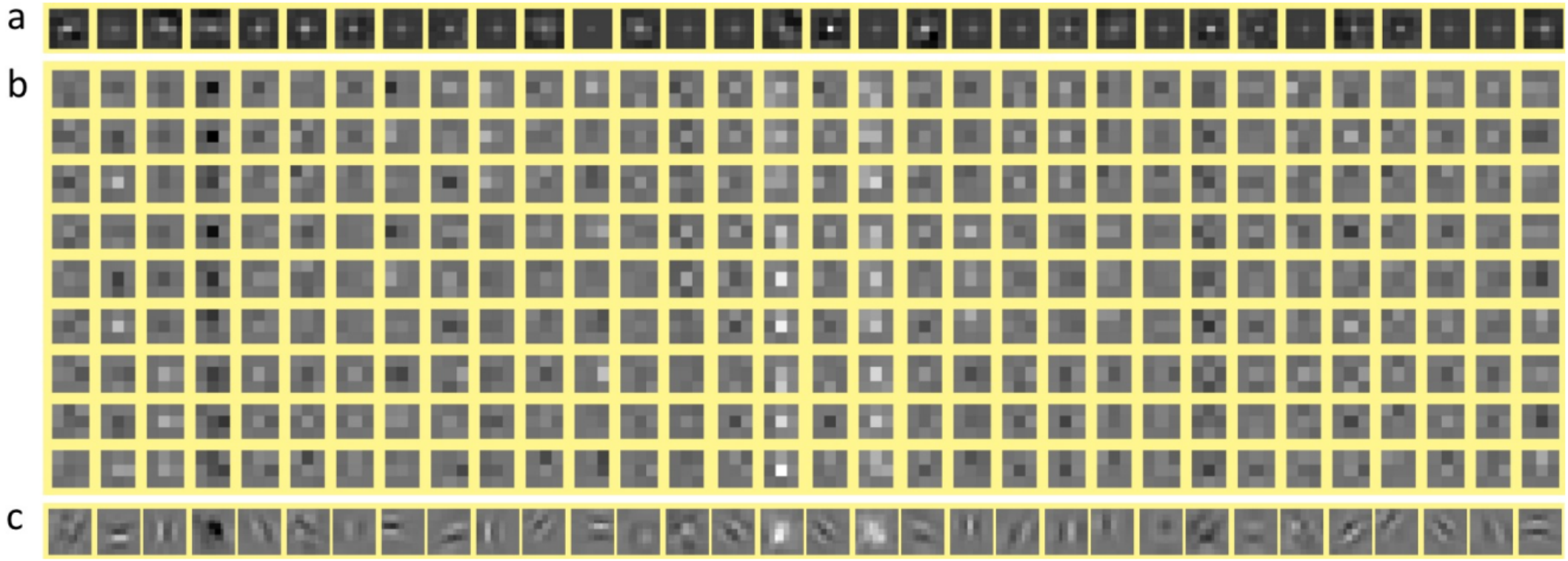


Figure 4. The last-layer filters trained on ImageNet with an upscaling factor of 3: (a) shows weights from SRCNN 9-5-5 model [7], (b) shows weights from ESPCN (ImageNet $relu$) model and (c) weights from (b) after the $\mathcal{PS}$ operation applied to the r^2 channels. The filters are in their default ordering.

Measurements

1. PSNR: Peak Signal to Noise Ratio
2. PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

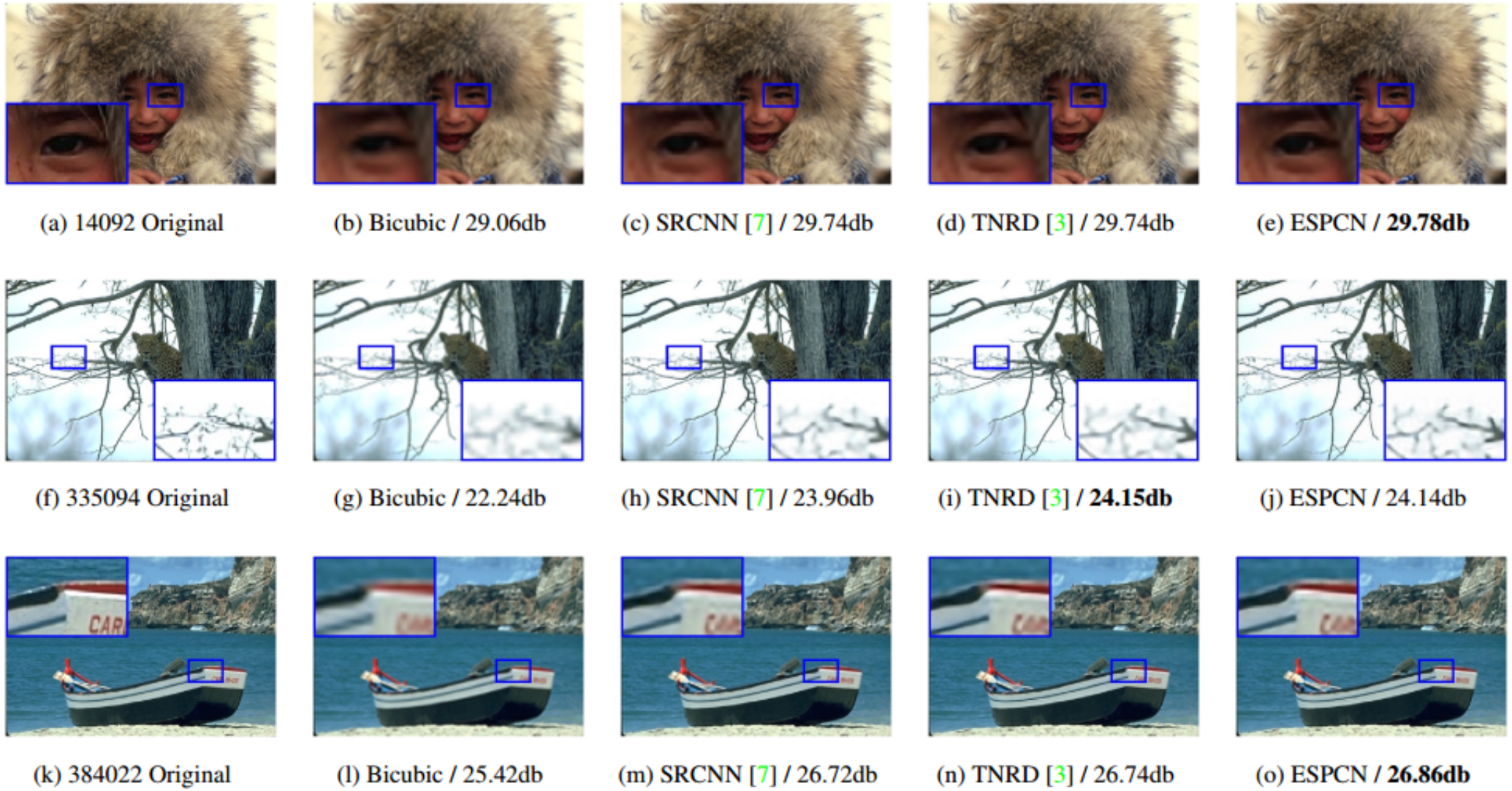
$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

(a) Baboon Original (b) Bicubic / 23.21db (c) SRCNN [7] / 23.67db (d) TNRD [3] / 23.62db (e) ESPCN / **23.72db**

(f) Comic Original (g) Bicubic / 23.12db (h) SRCNN [7] / 24.56db (i) TNRD [3] / 24.68db (j) ESPCN / **24.82db**

(k) Monarch Original (l) Bicubic / 29.43db (m) SRCNN [7] / 32.81db (n) TNRD [3] / 33.62db (o) ESPCN / **33.66db**

Figure 5. Super-resolution examples for ”Baboon”, ”Comic” and ”Monarch” from **Set14** with an upscaling factor of 3. PSNR values are shown under each sub-figure.



(a) 14092 Original (b) Bicubic / 29.06db (c) SRCNN [7] / 29.74db (d) TNRD [3] / 29.74db (e) ESPCN / **29.78db**

(f) 335094 Original (g) Bicubic / 22.24db (h) SRCNN [7] / 23.96db (i) TNRD [3] / **24.15db** (j) ESPCN / 24.14db

(k) 384022 Original (l) Bicubic / 25.42db (m) SRCNN [7] / 26.72db (n) TNRD [3] / 26.74db (o) ESPCN / **26.86db**

Figure 6. Super-resolution examples for ”14092”, ”335094” and ”384022” from **BSD500** with an upscaling factor of 3. PSNR values are shown under each sub-figure.

Activation comparison

Dataset	Scale	SRCNN (91)	ESPCN (91 $relu$)	ESPCN (91)	SRCNN (ImageNet)	ESPCN (ImageNet $relu$)
Set5	3	32.39	32.39	32.55	32.52	**33.00**
Set14	3	29.00	28.97	29.08	29.14	**29.42**
BSD300	3	28.21	28.20	28.26	28.29	**28.52**
BSD500	3	28.28	28.27	28.34	28.37	**28.62**
SuperTexture	3	26.37	26.38	26.42	26.41	**26.69**
Average	3	27.76	27.76	27.82	27.83	**28.09**

Table 1. The mean PSNR (dB) for different models. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and other methods (p-value < 0.001 with paired t-test).

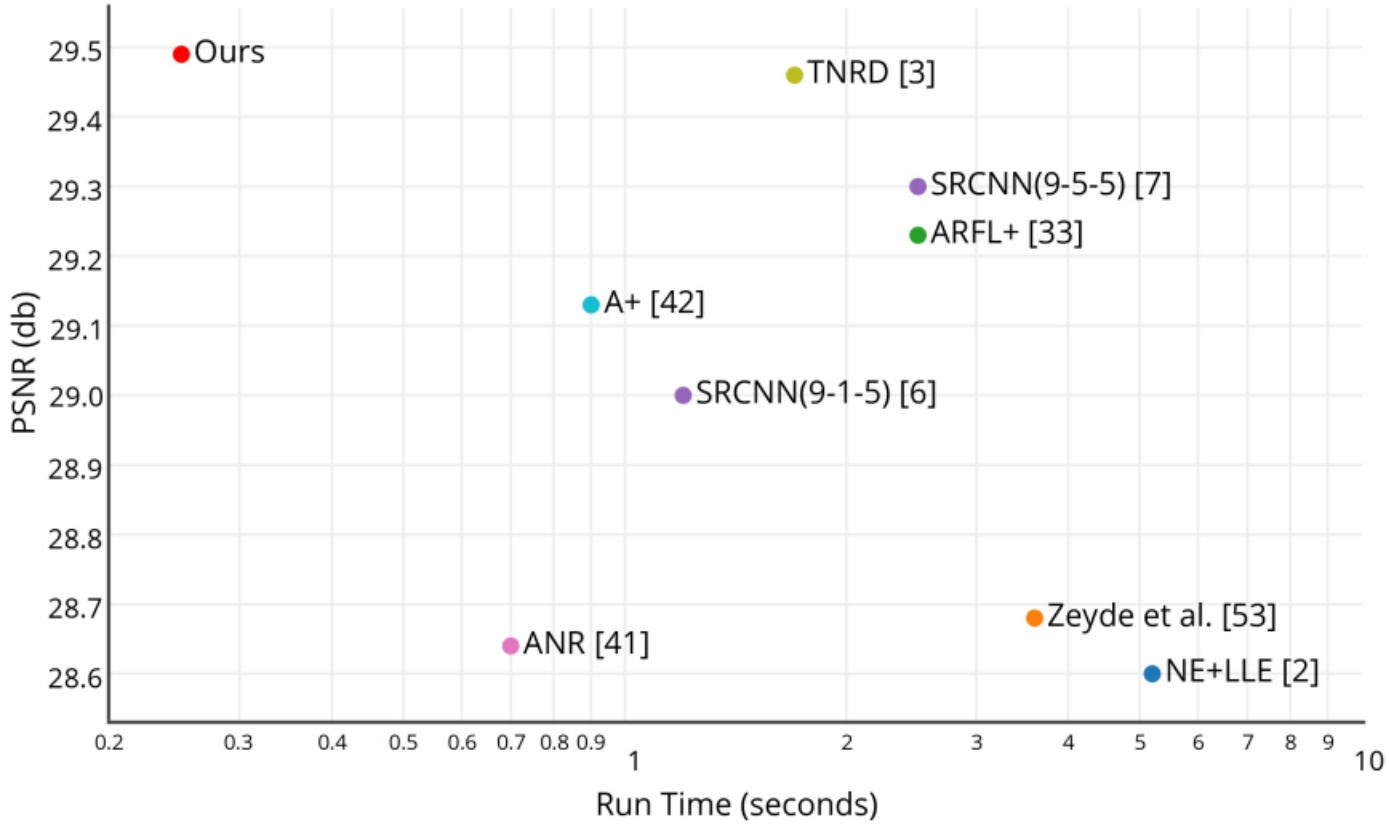


Figure 2. Plot of the trade-off between accuracy and speed for different methods when performing SR upscaling with a scale factor of 3. The results presents the mean PSNR and run-time over the images from Set14 run on a single CPU core clocked at 2.0 GHz.

Dataset	Scale	Bicubic	SRCNN	TNRD	ESPCN
Set5	3	30.39	32.75	**33.17**	33.13
Set14	3	27.54	29.30	29.46	**29.49**
BSD300	3	27.21	28.41	28.50	**28.54**
BSD500	3	27.26	28.48	28.59	**28.64**
SuperTexture	3	25.40	26.60	26.66	**26.70**
Average	3	26.74	27.98	28.07	**28.11**
Set5	4	28.42	30.49	30.85	**30.90**
Set14	4	26.00	27.50	27.68	**27.73**
BSD300	4	25.96	26.90	27.00	**27.06**
BSD500	4	25.97	26.92	27.00	**27.07**
SuperTexture	4	23.97	24.93	24.95	**25.07**
Average	4	25.40	26.38	26.45	**26.53**

Table 2. The mean PSNR (dB) of different methods evaluated on our extended benchmark set. Where SRCNN stands for the SRCNN 9-5-5 ImageNet model [7], TNRD stands for the Trainable Nonlinear Reaction Diffusion Model from [3] and ESPCN stands for our ImageNet model with $tanh$ activation. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and SRCNN (p-value < 0.01 with paired ttest)

Dataset	Scale	Bicubic	SRCNN	ESPCN
SunFlower	3	41.72	43.29	**43.36**
Station2	3	36.42	38.17	**38.32**
PedestrianArea	3	37.65	39.21	**39.27**
SnowMnt	3	26.00	27.23	**27.20**
Aspen	3	32.75	**34.65**	34.61
OldTownCross	3	31.20	32.44	**32.53**
DucksTakeOff	3	26.71	27.66	**27.69**
CrowdRun	3	26.87	28.26	**28.39**
Average	3	32.41	33.86	**33.92**
SunFlower	4	38.99	40.57	**41.00**
Station2	4	34.13	35.72	**35.91**
PedestrianArea	4	35.49	36.94	**36.94**
SnowMnt	4	24.14	24.87	**25.13**
Aspen	4	30.06	31.51	**31.83**
OldTownCross	4	29.49	30.43	**30.54**
DucksTakeOff	4	24.85	25.44	**25.64**
CrowdRun	4	25.21	26.24	**26.40**
Average	4	30.30	31.47	**31.67**

Table 3. Results on HD videos from **Xiph** database. Where SRCNN stands for the SRCNN 9-5-5 ImageNet model [7] and ESPCN stands for our ImageNet model with $tanh$ activation. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and SRCNN (p-value < 0.01 with paired t-test)

Dataset	Scale	Bicubic	SRCNN	ESPCN
Bosphorus	3	39.38	41.07	**41.25**
ReadySetGo	3	34.64	37.33	**37.37**
Beauty	3	39.77	40.46	**40.54**
YachtRide	3	34.51	36.07	**36.18**
ShakeNDry	3	38.79	40.26	**40.47**
HoneyBee	3	40.97	42.66	**42.89**
Jockey	3	41.86	43.62	**43.73**
Average	3	38.56	40.21	**40.35**
Bosphorus	4	36.47	37.53	**38.06**
ReadySetGo	4	31.69	33.69	**34.22**
Beauty	4	38.79	39.48	**39.60**
YachtRide	4	32.16	33.17	**33.59**
ShakeNDry	4	35.68	36.68	**37.11**
HoneyBee	4	38.76	40.51	**40.87**
Jockey	4	39.85	41.55	**41.92**
Average	4	36.20	37.52	**37.91**

Table 4. Results on HD videos from **Ultra Video Group** database. Where SRCNN stands for the SRCNN 9-5-5 ImageNet model [7] and ESPCN stands for our ImageNet model with $tanh$ activation. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and SRCNN (p-value < 0.01 with paired t-test)

Conclusions

1. In this paper, they propose to perform the feature extraction stages in the LR space instead of HR space
2. A novel sub-pixel convolution layer which is capable of super-resolving LR data into HR space with very little additional computational cost

Future work

1. Most of a scene's content is shared by neighboring video frames.
2. This creates additional data implicit redundancy that can be exploited for video super resolution.