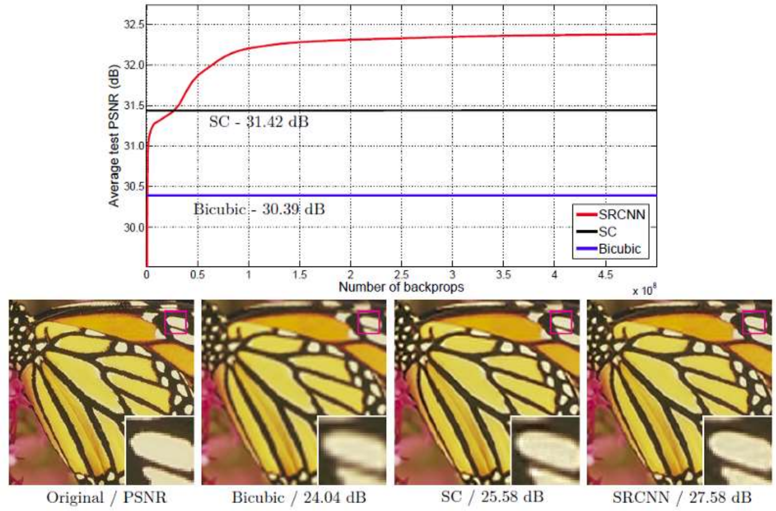
Review: SRCNN (Super Resolution)

SH Tsang Follow
Sep 5, 2018 · 5 min read

In this story, a very classical super resolution technique, **Super-Resolution Convolutional Neural Network (SRCNN)** [1–2], is reviewed. In deep learning or convolutional neural network (CNN), we usually use CNN for image classification. In SRCNN, it is used for **single image super resolution (SR)** which is a classical problem in computer vision.

In brief, with better SR approach, we can get a better quality of a larger image even we only get a small image originally.



Original / PSNR

Bicubic / 24.04 dB

SC / 25.58 dB

SRCNN / 27.58 dB

SRCNN

We can see from the above figure that, with SRCNN, PSNR of 27.58 dB is obtained which is much better than the classical non-learning based Bicubic and sparse coding (SC) which was and still is also a very hot research topic.

SRCNN is published in **2014 ECCV** [1] and **2016 TPAMI** [2] papers with both about **1000 citations** when I was writing this story. (SH Tsang @ Medium)

. . .

What are covered

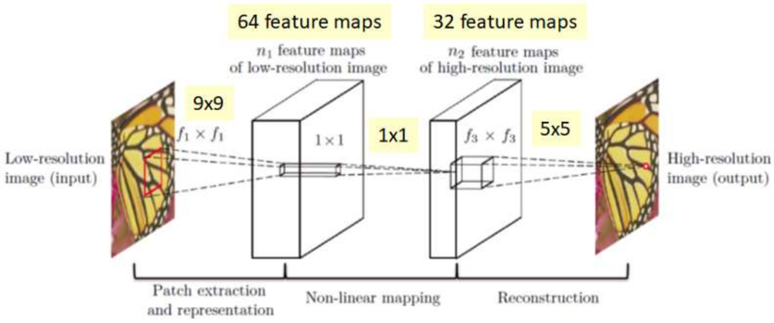
1. **The SRCNN Network**
2. **Loss Function**
3. **Relationship with Sparse Coding**
4. **Comparison with State-of-the-art Approaches**

5. **Ablation Study**

· · ·

1. The SRCNN Network

In SRCNN, actually the network is not deep. There are only 3 parts, patch extraction and representation, non-linear mapping, and reconstruction as shown in the figure below:


64 feature maps
32 feature maps
n_1 feature maps of low-resolution image
n_2 feature maps of high-resolution image
9x9
$f_1 \times f_1$
1×1
1x1
$f_3 \times f_3$
5x5
Low-resolution image (input)
High-resolution image (output)
Patch extraction and representation
Non-linear mapping
Reconstruction

SRCNN Network

1.1 Patch Extraction and Representation

It is important to know that **the low-resolution input is first upscale to the desired size using bicubic interpolation** before inputting to SRCNN network. Thus,
X: Ground truth high-resolution image
Y: Bicubic upsampled version of low-resolution image

And the first layer perform a standard conv with Relu to get F1(Y).

$$F_1(\mathbf{Y}) = \max\left(0, W_1 * \mathbf{Y} + B_1\right)$$

The first Layer

Size of W1: c×f1×f1×n1
Size of B1: n1

where c is number of channels of the image, f1 is the filter size, and n1 is the number of filters. B1 is the n1-dimensional bias vector which is just used for increasing the degree of freedom by 1.

In this case, **c=1, f1=9, n1=64**.

1.2 Non-Linear Mapping

After that, a non-linear mapping is performed.

$$F_2(\mathbf{Y}) = \max\left(0, W_2 * F_1(\mathbf{Y}) + B_2\right)$$

The second layer

Size of W2: n1×1×1×n2
Size of B2: n2

It is a mapping of n1-dimensional vector to n2-dimensional vector. When n1>n2, we can imagine something like PCA stuffs but in a non-linear way.

In this case, **n2=32.**

This 1×1 actually is a 1×1 convolution suggested in Network In Network (NIN) [3] as well. In NIN, 1×1 convolution is suggested to introduce more non-linearlity to improve the accuracy. It is also suggested in GoogLeNet [4] for reducing the number of connections. (Please visit my review for 1×1 convolution in GoogLeNet if interested.)

Here, it is used for mapping low-resolution vector to high-resolution vector.

1.3 Reconstruction

After mapping, we need to reconstruct the image. Hence, we do conv again.

$$F(\mathbf{Y}) = W_3 * F_2(\mathbf{Y}) + B_3$$

The third layer

Size of W3: n2×1 ×1×c
Size of B3: c

. . .

2. Loss Function

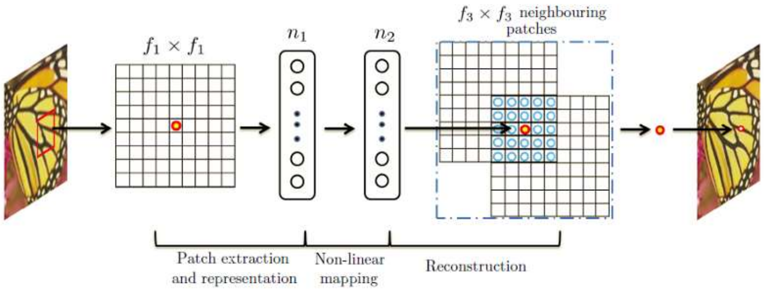
$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n} ||F(\mathbf{Y}_i;\Theta) - \mathbf{X}_i||^2$$

Loss function

For super resolution, the loss function *L* is the average of mean square error (MSE) for the training samples (n), which is a kind of standard loss function.

. . .

3. Relationship with Sparse Coding


$f_1 \times f_1$
n_1
n_2
$f_3 \times f_3$ neighbouring patches
Patch extraction and representation
Non-linear mapping
Reconstruction

Sparse Coding

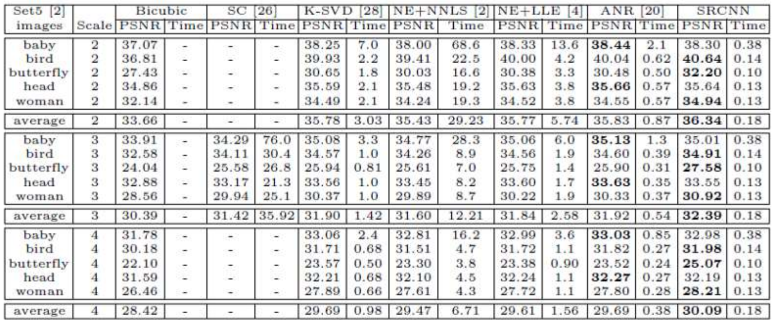
For Sparse Coding (SC), in the view of convolution, the input image is conv by f1 and project to onto a n1-dimensional dictionary. n1=n2 usually is the case of SC. Then mapping of n1 to n2 is done with the same dimensionality without reduction. It is just like a mapping of low-resolution vector to high-resolution vector. Then each patch is reconstructed by f3. And overlapping patches are averaged instead of adding together with different weights by convolution.

. . .

4. Comparison with State-of-the-art Approaches

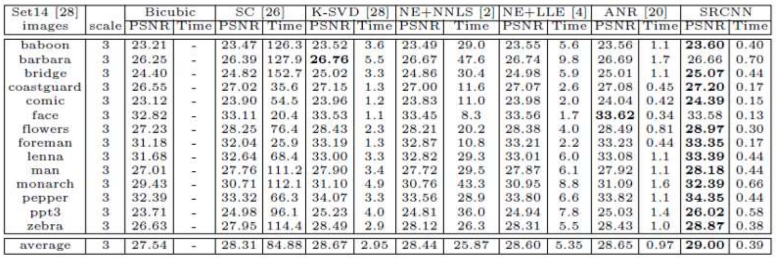
91 training images provide roughly 24,800 sub-images with stride 14 and Gaussian blurring. And takes 3 days for training on a GTX 770 GPU with 8×10^8 backpropagations.

Different scales from 2 to 4 are tested.



Set5 [2]		Bicubic		SC [26]		K-SVD [28]		NE+NNLS [2]		NE+LLE [4]		ANR [20]		SRCNN	
images	Scale	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
baby	2	37.07	-	-	-	38.25	7.0	38.00	68.6	38.33	13.6	**38.44**	2.1	38.30	0.38
bird	2	36.81	-	-	-	39.93	2.2	39.41	22.5	40.00	4.2	40.04	0.62	**40.64**	0.14
butterfly	2	27.43	-	-	-	30.65	1.8	30.03	16.6	30.38	3.3	30.48	0.50	**32.20**	0.10
head	2	34.86	-	-	-	35.59	2.1	35.48	19.2	35.63	3.8	**35.66**	0.57	35.64	0.13
woman	2	32.14	-	-	-	34.49	2.1	34.24	19.3	34.52	3.8	34.55	0.57	**34.94**	0.13
average	2	33.66	-	-	-	35.78	3.03	35.43	29.23	35.77	5.74	35.83	0.87	**36.34**	0.18
baby	3	33.91	-	34.29	76.0	35.08	3.3	34.77	28.3	35.06	6.0	**35.13**	1.3	35.01	0.38
bird	3	32.58	-	34.11	30.4	34.57	1.0	34.26	8.9	34.56	1.9	34.60	0.39	**34.91**	0.14
butterfly	3	24.04	-	25.58	26.8	25.94	0.81	25.61	7.0	25.75	1.4	25.90	0.31	**27.58**	0.10
head	3	32.88	-	33.17	21.3	33.56	1.0	33.45	8.2	33.60	1.7	**33.63**	0.35	33.55	0.13
woman	3	28.56	-	29.94	25.1	30.37	1.0	29.89	8.7	30.22	1.9	30.33	0.37	**30.92**	0.13
average	3	30.39	-	31.42	35.92	31.90	1.42	31.60	12.21	31.84	2.58	31.92	0.54	**32.39**	0.18
baby	4	31.78	-	-	-	33.06	2.4	32.81	16.2	32.99	3.6	**33.03**	0.85	32.98	0.38
bird	4	30.18	-	-	-	31.71	0.68	31.51	4.7	31.72	1.1	31.82	0.27	**31.98**	0.14
butterfly	4	22.10	-	-	-	23.57	0.50	23.30	3.8	23.38	0.90	23.52	0.24	**25.07**	0.10
head	4	31.59	-	-	-	32.21	0.68	32.10	4.5	32.24	1.1	**32.27**	0.27	32.19	0.13
woman	4	26.46	-	-	-	27.89	0.66	27.61	4.3	27.72	1.1	27.80	0.28	**28.21**	0.13
average	4	28.42	-	-	-	29.69	0.98	29.47	6.71	29.61	1.56	29.69	0.38	**30.09**	0.18

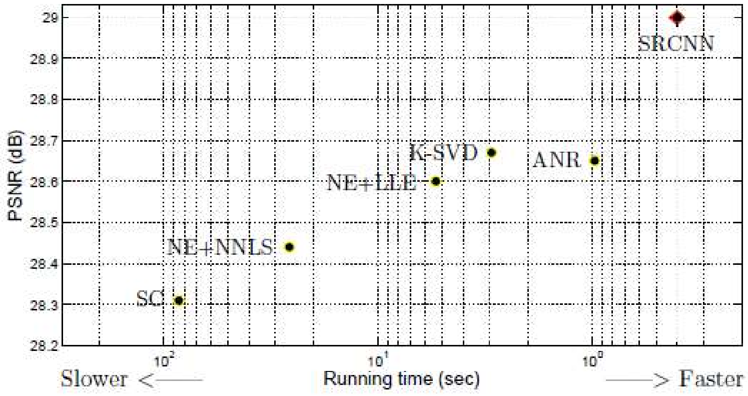
PSNR for Set15 dataset



Set14 [28]		Bicubic		SC [26]		K-SVD [28]		NE+NNLS [2]		NE+LLE [4]		ANR [20]		SRCNN	
images	scale	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
baboon	3	23.21	-	23.47	126.3	23.52	3.6	23.49	29.0	23.55	5.6	23.56	1.1	**23.60**	0.40
barbara	3	26.25	-	26.39	127.9	**26.76**	5.5	26.67	47.6	26.74	9.8	26.69	1.7	26.66	0.70
bridge	3	24.40	-	24.82	152.7	25.02	3.3	24.86	30.4	24.98	5.9	25.01	1.1	**25.07**	0.44
coastguard	3	26.55	-	27.02	35.6	27.15	1.3	27.00	11.6	27.07	2.6	27.08	0.45	**27.20**	0.17
comic	3	23.12	-	23.90	54.5	23.96	1.2	23.83	11.0	23.98	2.0	24.04	0.42	**24.39**	0.15
face	3	32.82	-	33.11	20.4	33.53	1.1	33.45	8.3	33.56	1.7	**33.62**	0.34	33.58	0.13
flowers	3	27.23	-	28.25	76.4	28.43	2.3	28.21	20.2	28.38	4.0	28.49	0.81	**28.97**	0.30
foreman	3	31.18	-	32.04	25.9	33.19	1.3	32.87	10.8	33.21	2.2	33.23	0.44	**33.35**	0.17
lenna	3	31.68	-	32.64	68.4	33.00	3.3	32.82	29.3	33.01	6.0	33.08	1.1	**33.39**	0.44
man	3	27.01	-	27.76	111.2	27.90	3.4	27.72	29.5	27.87	6.1	27.92	1.1	**28.18**	0.44
monarch	3	29.43	-	30.71	112.1	31.10	4.9	30.76	43.3	30.95	8.8	31.09	1.6	**32.39**	0.66
pepper	3	32.39	-	33.32	66.3	34.07	3.3	33.56	28.9	33.80	6.6	33.82	1.1	**34.35**	0.44
ppt3	3	23.71	-	24.98	96.1	25.23	4.0	24.81	36.0	24.94	7.8	25.03	1.4	**26.02**	0.58
zebra	3	26.63	-	27.95	114.4	28.49	2.9	28.12	26.3	28.31	5.5	28.43	1.0	**28.87**	0.38
average	3	27.54	-	28.31	84.88	28.67	2.95	28.44	25.87	28.60	5.35	28.65	0.97	**29.00**	0.39

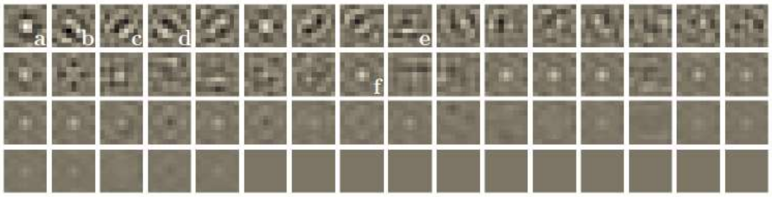
PSNR for Set14 dataset

SRCNN obtains the highest average PSNR.



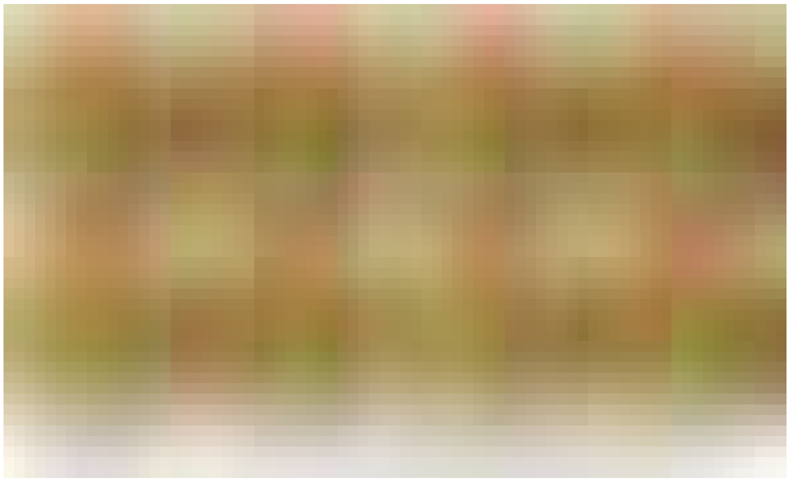
PSNR against Time

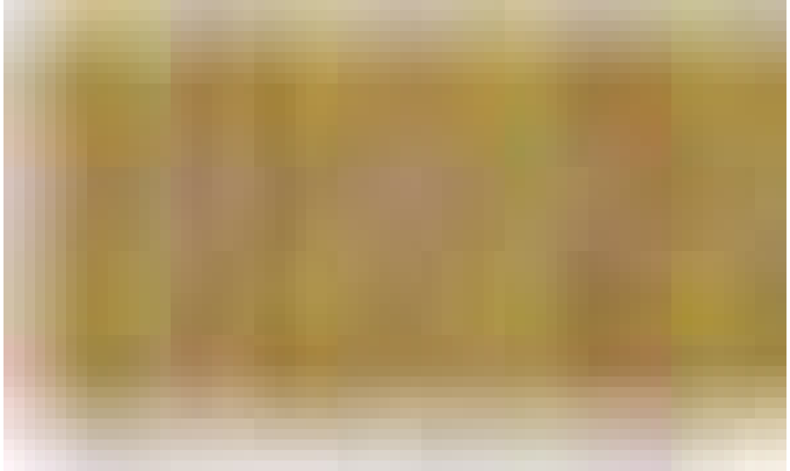
The righter, the faster, the higher, the better quality.
And **SRCNN is at the top right corner** which has the best performance.

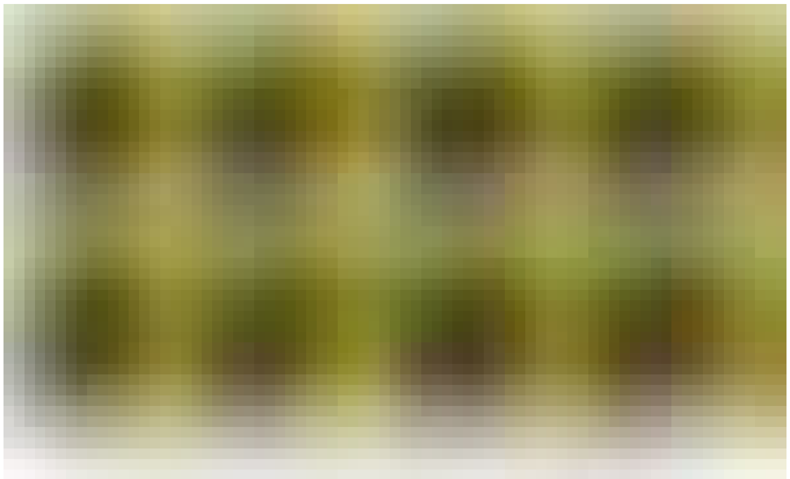


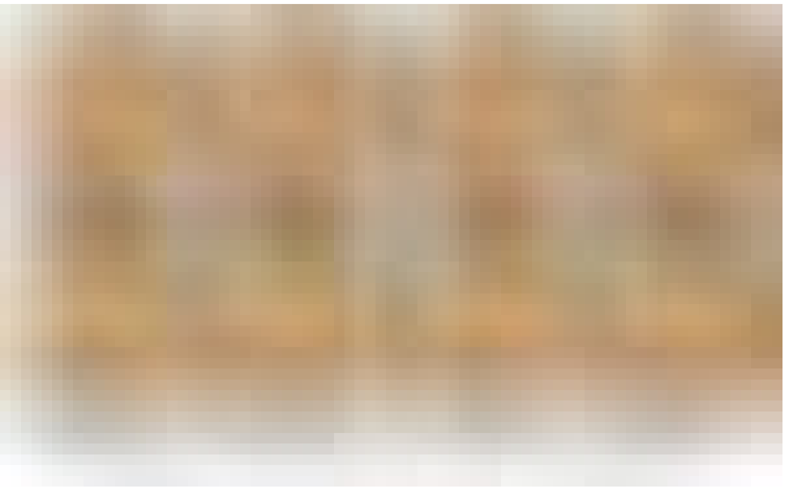
Visualization of first-layer filters

Some visual qualities:

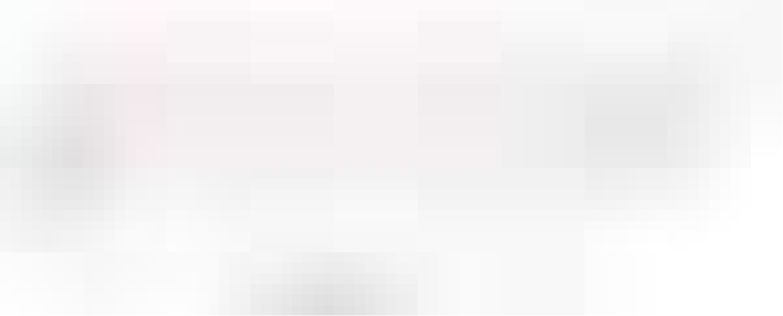






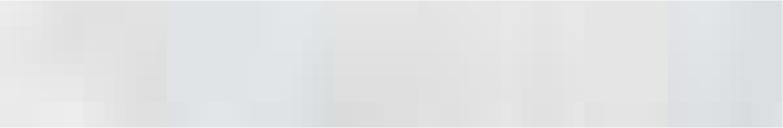


5. Ablation Study



Training from ImageNet vs Training from 91 images

If **SRCNN trained using 395,909 images** which is partially **from ILSVRC 2013 ImageNet** detection training dataset, the result is **better than just trained from 91 images.**



Different number of n1 and n2, Trained from ImageNet and Test on Set5

The larger n1 and n2, the higher the PSNR. It is normal as more filters, it should be better.

Also, with larger filter size, it also leads to a little better results. (But actually, there are only 3 layers, it is not sufficient enough to prove this. They should increase the layers as well. If there are more layers, larger filters can be replaced by several small filters.)

. . .

SRCNN contains only 3 layers. It is a easy and worth to read paper. So, it is also a paper to act as a starting point for learning deep learning or CNN! :)

. . .

References

1. [2014 ECCV] [SRCNN]
   Learning a Deep Convolutional Network for Image Super-Resolution

2. [2016 TPAMI] [SRCNN]
   Image Super-Resolution Using Deep Convolutional Networks

3. [2014 ICLR] [NIN]
   Network in Network

4. [2015] [CVPR] [GoogLeNet]
   Going Deeper with Convolutions

My Reviews

1. Review: GoogLeNet (Inception v1)—Winner of ILSVRC 2014 (Image Classification)