LARGE RECEPTIVE FIELD CONVOLUTIONAL NEURAL NETWORK FOR IMAGE SUPER-RESOLUTION

Qiang Wang[1,2], *Huijie Fan*[1], *Yang Cong*[1], *Yandong Tang*[1]

[1] State Key Laboratory of Robotics, Shenyang Institute of Automation,
Chinese Academy of Science, Shenyang 110016, China
[2] Graduate University of the Chinese Academy of Science, Beijing 100049, China

ABSTRACT

This paper presents a new approach to Single Image Super Resolution (SISR), based upon Convolutional Neural Network (CNN). Although the SISR is ill-posed which can be seen as finding a non-linear mapping from a low to high-dimensional space. Deep learning techniques have been successfully applied in many areas of computer vision, including low-level image restoration and non-linear mapping problems. We consider the single image Super-Resolution (SR) problem as convolution operators and develop a CNN to capture the characteristics of Low-Resolution (LR) input image. We find that increasing the receptive field shows the improvement in accuracy. Our solution is to establish the connection between traditional optimization-based schemes and neural network architectures. In the paper a novel, separable structure is introduced as a reliable support for robust convolution against artifacts. Our proposed method performs better than existing methods in terms of accuracy and visual improvements in our results are easily noticeable.

Index Terms— Super resolution, Convolutional neural network, Receptive field, Multi-scale

1. INTRODUCTION

Single Image Super Resolution (SISR) [16, 2, 14, 8] is an important computer vision problem with many interesting applications, ranging from medical and astronomical imaging to law enforcement. The aim in SISR is to generate a visually pleasing high-resolution output from a single LR input image generating a plausible and visually pleasing High-Resolution (HR) output image. Although the problem is inherently ambiguous and ill-posed, previous research including interpolation, bicubic and neighbor embedding [2] methods interpolate the patch subspace. Sparse coding [17, 14, 15, 3] methods exploit internal similarities of the same image, or learn mapping functions from external low-and high-resolution exemplar pairs. These methods are often provided with abundant samples, but are challenged by the difficulties of effectively and compactly modeling the data. Recently, random forest [13] and CNN [4, 9] have also been used with large improvements in accuracy. Among them, Dong et al. [4] had demonstrated that Convolutional Neural Network (CNN) can be used to learn a mapping from LR to HR in an end-to-end manner. This method, termed as SRCNN, does not require any engineered features that are typically necessary in other methods and shows the state-of-the-art performance.

In this work, We use the CNN to learn the convolution operation to resolve the issues. In fact, it is non-trivial to find a proper network architecture for the problem. Previous image restored neural network [5, 7] cannot be directly adopted since SR may involve many neighboring pixels and result in a very complex energy function with nonlinear mapping. This makes parameter learning quite challenging.

Overall, the contributions of this work are mainly in two aspects:

1. It is often the case that information contained in a small patch is not sufficient for detail recovery (ill-posed). Our network using large receptive field takes a large image context into account. We utilize ample contextual information to learn more rubust mapping of the image regions.

2. We adopt a multi-scale method to obtain additional feature maps, which have been proven effective for image restore. Multi-scale features densely compute features of an input image at multiple spatial scales.

2. RELATED WORK

A category of SR approaches [2, 6, 14, 15, 13] learn a mapping between low/high-resolution patches. Most studies vary on how to learn a compact dictionary or manifold space to relate low/high-resolution patches, and on how representation schemes can be conducted in such spaces. In the pioneer work of Timofte et al. [14], the dictionaries are directly presented as low/high-resolution patch pairs. And the Nearest Neighbour (NN) of the input patch is found in the low-resolution space, with its corresponding high-resolution patch used for reconstruction. Bevilacqua et al. [2] introduce a manifold embedding technique as an alternative to the NN strategy. In Yang et al.s work [16] , the above NN cor-

respondence advances to a more sophisticated sparse coding formulation. This sparse-coding-based method [14, 15, 11] and its several improvements are among the state-of-the-art SR methods nowadays. In these methods, the patches are focus of the optimization; the patch extraction and aggregation steps are considered as pre/post-processing and handled separately.

Convolutional neural networks date back decades and have recently shown an explosive popularity par-tially due to its success in image classification [10]. Several factors are of central importance in this progress: (i) the efficient training implementation on modern powerful GPUs, (ii) the proposal of the Rectified Linear Unit (ReLU) which makes convergence much faster and still presents good quality, and (iii) the easy access to an abundance of data (like ImageNet) for training larger models. Our method also benefits from these progresses. There have been a few studies of using deep learning techniques for image resolution. The multi-layer perceptron (MLP), whose all layers are fully-connected (in contrast to convolutional), is applied for natural image resolution. More closely related to our work, the convolutional neural network [9] which adopts 20 weight layers (3×3 for each layer) is applied for natural image resolution. However, we argue that increasing receptive field significantly boosts performance. We successfully use large filter kernal layers (61×61 for layer) to extract input image features.

3. PROPOSED METHOD

For SR image reconstruction, we present a new structure that use a very Large Receptive Field (61×61) convolutional network to this issue. Fig. 2 shows the architecture of Large Receptive Field Net (LRFCNN). Layers and nonlinear activations are designed to implement following sequential operations for Image Super-Resolution.

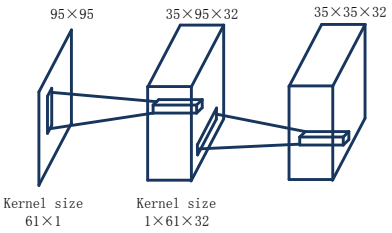


Fig. 1. Network architecture for large receptive convolution.

3.1. Feature extraction

Filters of spatial sizes 61×61 are used for feature extraction. However, simply modifying the network by employing large convolution kernels would lead to higher difficulties in training. We present a new structure to update the network as follows. We transform the large kernel for convolution based on the kernel separability theorem. It makes the network more expressive with the mapping to higher dimensions to accommodate nonlinearity. This system is benefited from large training data.

3.1.1. Kernel Separability

Kernel separability is achieved via Singular Value Decomposition (SVD) [1, 12]. Given the kernel $k^{\dagger}$, decomposition $k^{\dagger}$ = USV^T exists. We denote by u_j and v_j the jth columns of U and V ,s_j the jth singular value. Fig.1 shows the architecture of Large Receptive Field. The convolution can be expressed as

$$k^{\dagger} * y = \sum_i s_j \cdot u_j * (v_j^{\dagger} * y), \tag{1}$$

which shows 2D convolution can be deemed as a weighted sum of separable 1D filters. In practice, we can well approximate $k^{\dagger}$ by a small number of separable filters by dropping out kernels associated with zero or very smalls_j. We have experimented with real kernels to ignore singular values smaller than 0.01. The resulting average number of separable kernels is about 32. Using as smaller SNR ratio, the kernel has a smaller spatial support. We also found that an kernel with length 60 is typically enough to generate visually plausible convolution results. This is important information in designing the network architecture.

3.1.2. Multi-scale

Multi-scale has been proven effectively for regress net, which densely compute features of an input image at multiple spatial scales. Multi-scale is also effective to achieve scale invariance. Motivated by these successes of multi-scale, we choose to use parallel convolutional operations in the second layer of our CNN, where size of convolution filters is among 3×3, 5×5 and 7×7, and we use the same number of filters for these three scales. Formally, the output of the multi-scale layer is written as

$$F_2^i = \underset{j \in 1,2,3}{Concat}\{F_2^{i,j}\}, \quad F_2^{i,j} = W_2^{i,j} * F_1 + B_{i,j}\ , \tag{2}$$

where $W_2^{i,j}$and $B_2^{i,j}$ contain n_2 pairs of parameters that is break up into 3 groups. n_2 is the output dimension of the second layer, and i$\in [1, n_2\ ]$ indexes the output feature maps. The last convolutional block in Non-linear mapping is 192-d, and we attach a randomly initialized 64-d 1×1 convolutional layer for reducing dimension (to be precise, this increases the depth).

3.1.3. Non-linear mapping

The multi-scale layer extracts an n_1-dimensional feature for each patch. In the next operation, we map each of these n_1-dimensional vectors into an n_2-dimensional one. We also applys n_2 filters which have a trivial spatial support 1×1 to

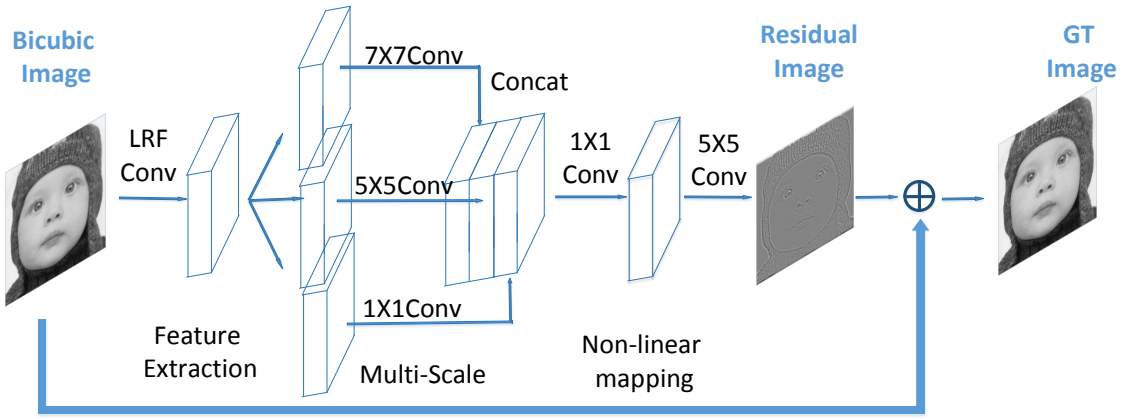


Fig. 2. Our complete network architecture for image super-resolution.

imply non-linear mapping. The operation of the layer is:

$$F_2(Y) = Max\{0, W_2 * F_1(Y) + B_2\} \quad (3)$$

Here W_2 is of a size $n_1 \times 1 \times 1 \times n_2$, and B_2 is n_2-dimensional

3.2. Reconstruction

In the Sparse-Coding-Based methods, the predicted overlapping high-resolution patches are often averaged to produce the final full image. David Eigen [5] illustrate the benefit of convolutional network, which is less correlated with one another compared dirct-averaging and produce a better performance. We also used a convolutional layer to produce the final high-resolution image:

$$F_3(Y) = W_3 * F_2(Y) + B_3 \quad (4)$$

Here, W_3 is the size of $n_2 \times f_3 \times f_3 \times c$, and B_3 is a c-dimensional vector. If the representations of the high-resolution patches are in the image domain, we expect that the filters act like an averaging filter; If the representations of the high-resolution patches are in some other domains, we expect that W_3 behaves like first projecting the coefficients onto the image domain and then averaging. In either way, W_3 is a set of linear filters. We put all operations together and form a convolutional neural network. In this model, all the filtering weights and biases are to be optimized.

3.3. Image Super-Resolution

Our complete network is formed as the CNN module with SRCNN. The overall structure is shown in Fig 2. The LRFCNN module has two hidden layers with 32 feature maps. The input image is convolved with 32 kernels of size 1×61 and 61×1 to be fed into the hidden layer. The following network modules are concatenated in our system by combining the last layer of LRFCNN with the input of Non-linear mapping layer. This is done by merging the 7×7 (5×5, 3×3) $\times$ 64 kernels to generate 64 kernels of size 35×35. Note that there is nonlinearity when combining the two modules. While the number of weights grows due to the merge, it allows for a flexible procedure and achieves decent performance, by further incorporating fine tuning.

4. EXPERIMENTAL RESULTS

For color images, we apply the proposed algorithm only on gray scale channel. In the training phase, the images T_i are prepared as 95×95-pixels sub-images randomly cropped from the training images and the ground thruth images X_i are the residual parts which define T_i subtracting the low-resolution parts. To synthesize the low-resolution samplesY_i, we blur a sub-images by a proper Gaussian kernel, subsample it by the upscaling factor, and upscale it by the same factor via bicubic interpolation. The network produces a smaller output (31×31). The MSE loss function is evaluated only by the difference between the central 31×31 crop of X_i.

4.1. Datasets for training and testing

We use 91 images from Yang et al. [17] and 200 images from the training set of Berkeley Segmentation Dataset as our training data. We perform experiments on three widely used benchmark datasets Set5 from [4] , Set14 from [4] , and B100 from [9] , consisting of 5, 14, and 100 images, respectively. In addition, data augmentation (rotation or flip) is used. For results in previous sections, we used 91 images to train network fast, so performances can be slightly different.

4.2. Training

We now describe the objective to find optimal parameters of our model. Let x denote an interpolated low-resolution image and y a high-resolution image. Given a training dataset (x_i, y_i), our goal is to learn a model f that predicts values $\hat{y}$ =

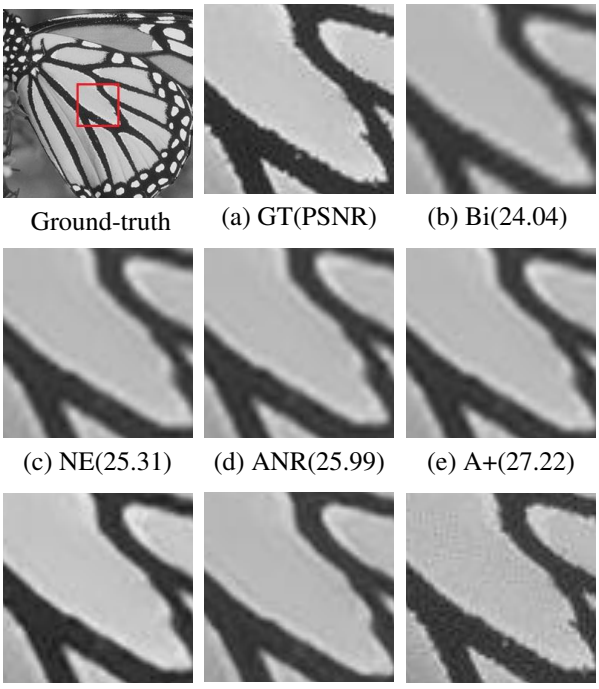
Datasets	Scale	Bicubic PSNR	ANR PSNR	A+ PSNR	RFL PSNR	SRCNN PSNR	LRCNN(Ours) PSNR
Set5	X2	33.66	35.87	36.55	36.54	36.66	**36.87**
	X3	30.29	31.91	32.59	32.43	32.75	**33.26**
	X4	28.37	29.73	30.28	30.14	30.48	**31.05**
Set14	X2	30.24	31.78	32.28	32.26	32.28	**33.03**
	X3	27.55	28.68	29.13	29.05	**29.28**	29.22
	X4	26.00	26.91	27.32	27.24	27.29	**27.31**
B100	X2	29.56	30.94	31.23	31.16	31.36	**31.69**
	X3	27.21	27.95	28.31	28.22	28.41	**28.82**
	X4	25.92	26.56	26.82	26.75	26.90	**27.09**

Table 1. Average PSNR for scale factor 2, 3 and 4 on datasets Set5, Set14, B100.

$f(x)$, where ŷ is an estimate of the target HR image. We minimize the mean squared error $\frac{1}{2}\|\hat{y} - f(x)\|$ averaged over the training set is minimized.

4.2.1. Residual-Learning

We also adopted strategy proposed by Jiwon Kim [9] , which called Residual-Learning. As the input and output images are largely similar, Jiwon Kim define a residual image $r = y - x$, where most values are likely to be zero. The aim of networks is to predict this residual image. The loss function now becomes $\frac{1}{2}\|r - f(x)\|$, where $f(x)$ is the network prediction.



Ground-truth (a) GT(PSNR) (b) Bi(24.04)

(c) NE(25.31) (d) ANR(25.99) (e) A+(27.22)

(f) RFL(27.11) (g) SRC(27.58) (h) Ours(27.85)

Fig. 3. The comparison of our super-resolution approach (STN) with the state-of-the-arts in terms of PSNR (upscaling factor is 3).

4.2.2. Training parameters

We provide parameters used to train our final model. Training uses batches of size 64. Momentum and weight decay parameters are set to 0.9 and 0.0001, respectively. For weight initialization, we use random-weight initialization. We train all experiments over 90 epochs (10000 iterations with batch size 64). Learning rate was initially set to 0.01 and then decreased by a factor of 10 every 20 epochs. In total, the learning rate was decreased 5 times, and the learning is stopped after 90 epochs. Training takes roughly 2 days on GPU K40.

4.3. Comparisons with state-of-the-art methods

We provide quantitative and qualitative comparisons. Compared methods are A+ [15], ANR[14] ,RFL [13] and SRCNN [4]. In Table 1, we provide a summary of quantitative evaluation on several datasets. Our method most outperform all previous methods in these datasets.

5. CONCLUSION

In this work, we present a super-resolution method using very large receptive filed networks. Training a very large network is much time-consuming due to a large convolutional filter. We use residual-learning and kernel keparability to optimize the very large network fast. Convergence speed is maximized and we use gradient clipping to ensure the training stability. We demonstrat that our method outperforms the existing methods by a large margin on benchmarked images. We believe our approach is readily applicable to other image restoration problems such as denoising and compression artifact removal.

Acknowledgements

This work has been funded by Natural Science Foundation of China(Grant No.61401455)

6. REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing over-complete dictionaries for sparse representation. In *TSP*, volume 54, pages 4311–4322, 2006.

[2] M. Bevilacqua, A. Roumy, C. Guillemot, A. Morel, M. Bevilacqua, and A. Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. *BMVC*, 2012.

[3] C. Dang and H. Radha. Fast image super-resolution via selective manifold learning of high-resolution patches. In *ICIP*, pages 1319–1323, 2015.

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *PAMI*, 38(2):295–307, 2016.

[5] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *IEEE International Conference on Computer Vision*, pages 633–640, 2013.

[6] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ECCV*, pages 349–356, 2009.

[7] S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, pages 2392–2399, 2012.

[8] J. B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015.

[9] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. *CVPR*, 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25(2):2012, 2012.

[11] H. S. Mousavi and V. Monga. Sparsity based super resolution using color channel constraints. In *ICIP*, pages 579–583, 2016.

[12] P. Perona. Deformable kernels for early vision. *PAMI*, 17(5):488–499, 1995.

[13] S. Schulter, C. Leistner, and H. Bischof. Fast and accurate image upscaling with super-resolution forests. In *CVPR*, 2015.

[14] R. Timofte, V. De, and L. V. Gool. Anchored neighborhood regression for fast example-based super-resolution. In *ECCV*, pages 1920–1927, 2013.

[15] R. Timofte, V. D. Smet, and L. V. Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2014.

[16] C. Y. Yang and M. H. Yang. Fast direct super-resolution by simple functions. In *ICCV*, pages 561–568, 2013.

[17] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *TIPS*, 19(11):2861–2873, 2010.