A HYBRID WAVELET CONVOLUTION NETWORK WITH SPARSE-CODING FOR IMAGE SUPER-RESOLUTION

Xing Gao *Hongkai Xiong*[*]

Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ABSTRACT

This paper proposes a hybrid wavelet convolution network (HWCN) which is composed of a scattering convolution component and a convolution neural component. The hierarchical end-to-end network implements sparse-coding and high-dimensional reconstruction for inverse problem through cascade convolutions. With the pre-defined scattering convolutions from nonlinear operators, the network can be tailored in accordance with the frequency property to provide sparse code candidates, and the convolution neural component could automatically select and weight these candidates for sparse coding. Given a tiny dataset, HWCN could train complex deep network with better generalization by regularization from scattering convolutions, and thereby is a competitive alternative to convolutional neural networks (CNN). Moreover, we further demonstrate that HWCN is a superior selection of sparse-coding based image super-resolution and achieves state-of-the-art performance.

***Index Terms*—** Scattering convolution network, convolutional neural network, sparse coding, image super-resolution

1. INTRODUCTION

Sparse representation, seeking to describe a signal as a linear combination of a few atoms from a pre-specified dictionary, presents an extraordinary power in a wide range of applications. In terms of the dictionary, it can be obtained by selecting an analytic dictionary based on a mathematical model or learning a dictionary from dataset which usually achieves state-of-the-art performance. For learned dictionary, however, during training phase, you need to optimize dictionary and sparse coding alternatively. Recently, convolutional neural networks (CNN) become prevailing in a series of computer vision fields. In fact, the networks are not well understood because of cascaded nonlinearities. To well disclose the properties and optimal configurations of large-scale invariants to deformations, S. Mallat $et\ al.$ [1, 2] developed a special kind of convolution network called scattering convolution network. Although the spatial receptive fields of simple cells in mammalian striate cortex have been characterized as being localized, oriented, and bandpass, comparable with the basis functions of wavelet transforms, it limits the flexibility and adaptivity since all of filters are predefined complex wavelets with nonlinear modulus and averaging pooling functions.

Image super-resolution, aiming to restore a high-resolution (HR) image from one or serval low-resolution (LR) counterparts, is a well-known inverse problem. Inspired from compressed sensing [3, 4], sparse-coding based methods play a representative role in image super-resolution and achieve state-of-the-art performance [5–8]. Also, the example-based learning methods achieve significant performance, including manifold learning [9], locally linear regression [10], and deep convolutional network [11].

This paper designs a hierarchical end-to-end network, the hybrid wavelet convolution network (HWCN), where we tailor the scattering convolution component to provide sparse code candidates. The convolutional neural component could automatically select and weight these candidates to generate sparse code and reconstruct a high-dimensional signal. Specifically, the scattering convolution component can prune negligible scattering paths in terms of the frequency property of complex wavelet kernels, and the convolution neural component can complement adaptivity and flexibility by learning filters from data. It is worth mentioning that the scattering network introduces a kind of regularization so that it is shown HWCN can achieve better generalization with more complex and deeper structure, even trained with a tiny dataset. Thereby, it is a competitive alternative to CNN and is easy to realize and improve performance in a hierarchical end-to-end sense.

2. HYBRID WAVELET CONVOLUTION NETWORK

As illustrated in Fig. 1, the scattering convolution component in HWCN provides a set of sparse coding candidates over a directional wavelet frame, while the convolution neural component automatically selects and weights the coefficients of the LR and HR images, and reconstructs the HR image from the sparse coefficients, respectively.

2.1. The Scattering Convolution Component

The scattering convolution component is a scattering convolution network, a wavelet tree in nature, each branch (filter)

[*]The work has been partially supported by the NSFC grants 61425011, U1201255 and 61271218.

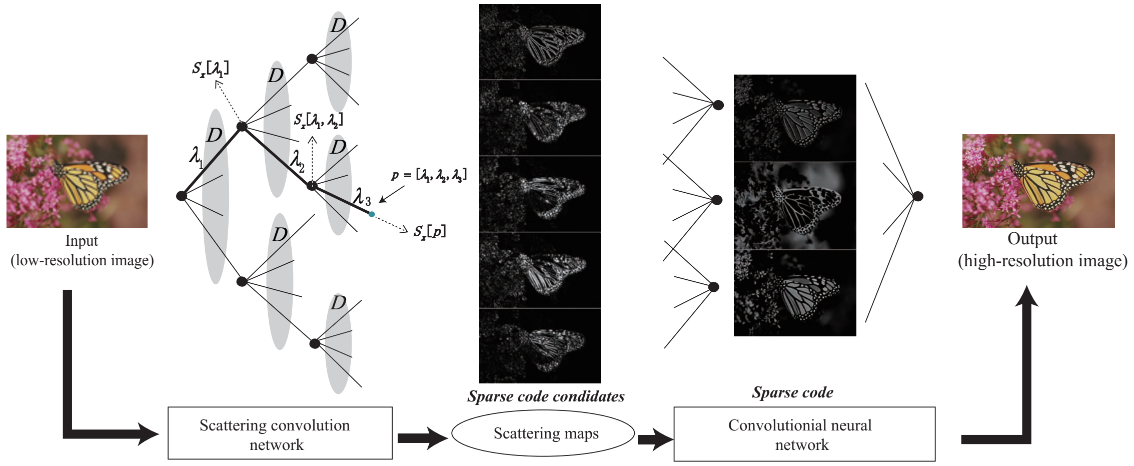


Fig. 1. The architecture of the hybrid wavelet convolution network. The low-resolution image is fed into the scattering convolution component to generate sparse code candidates, the first layer of the convolution neural component adaptively selects and weights these candidates to obtain sparse coefficients, and finally reconstruct the high-resolution image.

of which is a complex directional wavelet followed by a nonlinear modulus operator. All of the branches of a node form a wavelet defined dictionary, which is a common dictionary shared by all of the nodes, and the whole tree is an overcomplete dictionary, as illustrated in Fig. 1. Through the layer-by-layer convolutions, it produces wavelet coefficients, all of which form sparse code candidates called the scattering maps.

Two-dimensional complex directional wavelets with J scales and L directions are obtained by dilating and rotating a complex mother wavelet $\psi(u)$, and we define the set of these wavelets as dictionary D:

$$D = \{\psi_\lambda(u) : 2^{-2j}\psi(2^{-j}r^{-1}u)\}_{\lambda \in \Lambda}, \tag{1}$$

with

$$\Lambda = \{\lambda = 2^{-j}r | j = 0, 1, \cdots, J-1; \\ r = 0, 2\pi/L, \cdots, 2(L-1)\pi/L\}, \tag{2}$$

which can essentially be taken to form a semi-discrete shift-invariant Parseval frame [12]. Each wavelet in D defines a filter (branch) of each node of the scattering network. There is a path $p = [\lambda_1, \lambda_2, \cdots, \lambda_m]$ from the root node to each node, along which a scattering map is shaped by a series of convolution and modulo arithmetic:

$$S_x[p](u) = |||x * \psi_{\lambda 1}| * \psi_{\lambda 2}| \cdots * \psi_{\lambda m}|. \tag{3}$$

Scattering maps are sparse in nature due to the localization and orientation property of wavelets in D. Since modulus operator takes the envelope of signal and removes oscillations, it shifts signal into lower frequency so that we can only care about frequency decreasing paths($|\lambda_{i+1}| < |\lambda_i|$). All of the scattering maps of frequency decreasing paths form a sparse code candidate set:

$$S_x = \{S_x[p](u)\}_p. \tag{4}$$

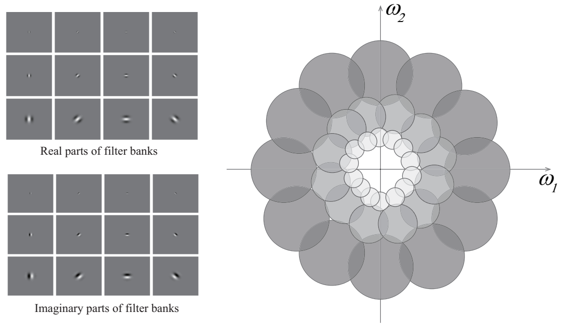


Fig. 2. Illustration of the real parts and imaginary parts of filter bank of the scattering convolutions in time domain, and the corresponding partition diagram of the frequency domain.

Each of these scattering maps captures distinct information of specific orientation and resolution. In frequency domain, wavelets with the same scale but different orientations in D nearly form an annulus, and all of these annuluses almost occupy the circle $|\omega| \leq \pi$, as illustrated in Fig. 2. According to the equivalence of convolution in time domain and multiplication in frequency domain, the spectrum of signal x has been divided into different sectors after the first layer. The following modulus operator shifts each sector into lower frequency, which is in accordance to our intuition that envelop changes more slowly. At the second layer, each shifted sector continues to be divided by these filters $\{\psi_\lambda\}_{\lambda \in \Lambda}$. With the increment of the number of layers, each sector is divided more elaborately. Overall, all of these scattering maps describe various information of different scales and orientations.

2.2. The Convolution Neural Component

The convolution neural component is a convolution neural network which first automatically select and weight candidates obtained from scattering convolutions to produce the sparse code and then reconstruct the HR image.

The first layer extracts common sparse code of the LR and HR image pairs. It has n_1 neurons, each of which takes all of the scattering maps as input, totally n_0 scattering maps, and employs the ReLU as the non-linearity operator. With $W_1(n_0 \times m_1 \times m_1 \times n_1)$ and B_1 $(n_1 \times 1)$ donoting filter bank and biases, the operation of the first layer is:

$$H_x = max(0, W_1 * S_x + B_1). \qquad (5)$$

The sparsity is achieved and guaranteed by three ingredients in the underlying mechanism. Above all, due to the localization and orientation characteristics of directional wavelets in the scattering convolutions, the sparse code candidates are sparse. In addition, the ReLU non-linearity operator in the first layer which omits negative ones makes these candidates further sparser. Finally, the number of output of the first layer n_1 is less than half of that of its input n_0, which forces each neuron of the first layer to select and weight these effective candidates so as to further ensure the sparsity.

The second layer employs the sparse code to reconstruct the HR image with only one neuron :

$$F_x = W_2 * H_x + B_2, \qquad (6)$$

where W_2 $(n_1 \times m_2 \times m_2 \times 1$) and B_2 denote filter bank and bias, respectively.

3. ANALYSIS

3.1. Relationship To CNNs

With respect to feature extraction, the scattering convolutions in HWCN can effectively extract multiscale and multidirectional features. For instance, the first, third, and fifth feature maps of CNN in Fig. 3 are the response of different orientations. However, some of the feature maps of CNN make no sense and even nearly contain nothing as shown in the forth map in Fig. 3. In contrast, features extracted by wavelets can be selected in the scattering convolutions so that to extract specific features, which is unrealizable for typical CNNs. In addition, the scattering convolution component of HWCN is predefined, which reduces the number of parameters required to train and provides a kind of regularization. Thus, we can use a tiny dataset to train complex deep networks with better generalization, as shown in Table 1 and Table 2, where the HWCN improves more than 0.1 dB PSNR than the deep CNN [11] for all scales.

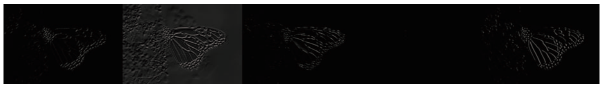


Fig. 3. Feature maps extracted by the first layer of CNN [11].

Table 1. Performance measured in PSNR (dB) on Set5 .

Set5	Scale	Bicubic	SC [6]	K-SVD [8]	NE+LLE [9]	ANR [10]	CNN [11]	Proposed
baby	2	37.005	-	38.186	38.276	**38.382**	38.238	38.327
bird	2	36.849	-	39.964	40.037	40.089	40.663	**41.056**
butterfly	2	27.446	-	30.708	30.449	30.548	32.310	**32.407**
head	2	34.804	-	35.545	35.584	35.610	35.589	**35.672**
woman	2	32.23	-	34.552	34.614	34.626	34.977	**35.340**
average	2	33.667	-	35.791	35.792	35.851	36.355	**36.561**
baby	3	33.870	34.258	35.038	35.018	**35.092**	34.968	34.903
bird	3	32.648	34.266	34.689	34.678	34.724	35.052	**35.504**
butterfly	3	24.064	25.675	26.000	25.809	25.964	**27.677**	27.648
head	3	32.824	33.150	33.508	33.556	33.590	33.499	**33.703**
woman	3	28.654	30.073	30.451	30.335	30.444	31.007	**31.260**
average	3	30.412	31.484	31.937	31.879	31.963	32.441	**32.604**
baby	4	31.752	-	**33.037**	32.964	33.008	32.955	32.868
bird	4	30.200	-	31.741	31.744	31.850	32.012	**32.358**
butterfly	4	22.129	-	23.614	23.422	23.565	**25.151**	24.798
head	4	31.548	-	32.177	32.203	32.230	32.149	**32.360**
woman	4	26.524	-	27.945	27.791	27.867	28.249	**28.718**
average	4	28.431	-	29.703	29.625	29.704	30.103	**30.220**

Table 2. Performance measured in PSNR (dB) and average time (s) on Set14.

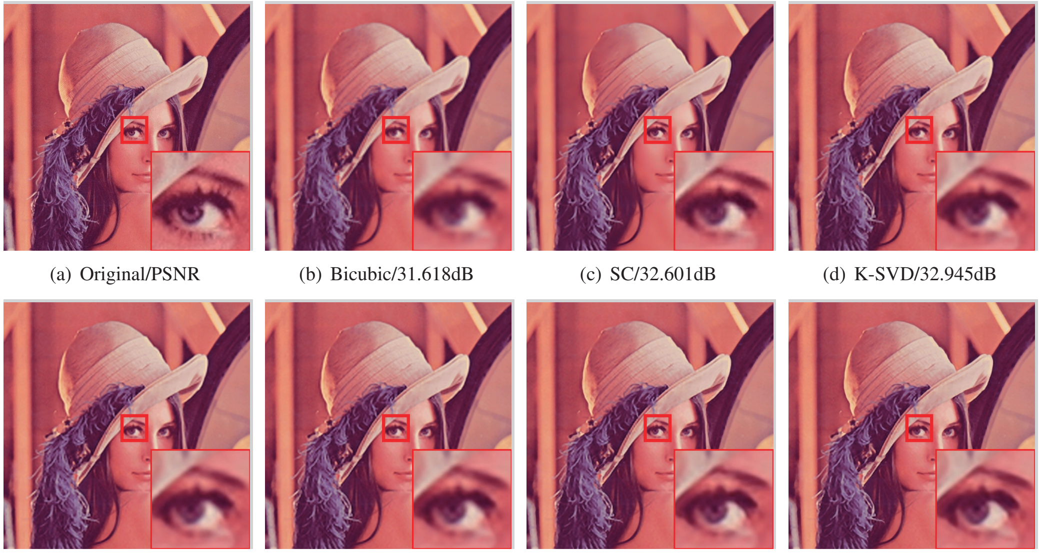
Set14	Scale	Bicubic	SC [6]	K-SVD [8]	NE+LLE [9]	ANR [10]	CNN [11]	Proposed
baboon	3	23.210	23.482	23.523	23.558	23.569	23.605	**23.617**
barbara	3	26.191	26.348	**26.699**	26.680	26.635	26.597	26.513
bridge	3	24.427	24.855	25.045	25.005	25.036	25.104	**25.195**
coastguard	3	26.715	27.054	27.168	27.146	27.174	27.196	**27.208**
comic	3	23.045	23.844	23.882	23.901	23.966	24.307	**24.405**
face	3	32.759	33.086	33.475	33.517	33.566	33.525	**33.710**
flowers	3	27.151	28.172	28.350	28.304	28.413	28.894	**29.020**
foreman	3	31.667	33.336	33.743	33.805	33.850	34.339	**34.759**
lenna	3	31.618	32.601	32.945	32.958	33.027	33.344	**33.472**
man	3	26.978	27.738	27.873	27.839	27.895	28.150	**28.302**
monarch	3	29.348	30.646	31.029	30.872	31.016	32.316	**32.349**
pepper	3	32.435	33.390	34.130	33.879	33.905	34.447	**34.720**
ppt3	3	23.619	24.890	25.136	24.853	24.939	25.933	**26.133**
zebra	3	26.563	27.898	28.435	28.254	28.371	28.818	**28.984**
average	3	27.552	28.381	28.674	28.612	28.669	29.041	**29.171**
avg time (s)	3	-	80.8284	3.8231	4.9020	0.7295	2.1742	3.6492

3.2. Relationship To Typical Dictionary Based Methods

Revisiting typical dictionary based methods, an HR image y is recovered from its LR counterpart x by the sparse code α under the HR and LR dictionary D_h, D_l:

$$y = D_h\alpha, \quad x = Py = PD_h\alpha = D_l\alpha. \qquad (7)$$

HWCN realizes such reconstruction with the scattering convolutions serving as an analytic LR dictionary and the convolution neural operations being an HR dictionary. While typical dictionary based methods optimize LR and HR dictionaries alternatively and seek for sparse code iteratively, we fix the LR dictionary and produce sparse code candidates directly, and then learn to select and weight sparse code and train the HR dictionary from data together. Moreover, the procedure is forward and end-to-end so that it can produce the HR image directly once the convolution neural component has been trained. As illustrated in Table 2, HWCN achieves more than 0.5 dB gain in terms of PSNR and takes less average time than typical dictionary based methods [6] [8].



(a) Original/PSNR (b) Bicubic/31.618dB (c) SC/32.601dB (d) K-SVD/32.945dB

(e) NE+LLE/32.958dB (f) ANR/33.027dB (g) CNN/33.344dB (h) Proposed/**33.472dB**

Fig. 5. “*Lenna*” image from Set14 for upscaling factor 3.

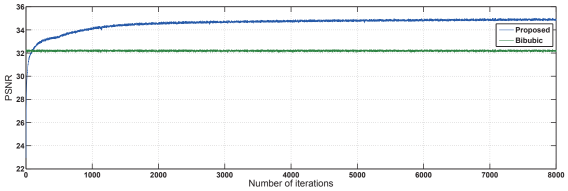


Fig. 4. The training convergence curve for upscaling factor 2.

4. EXPERIMENTS

Configuration For the scattering convolution component, we choose complex Morlet wavelet as mother wavelet and allocate $J = 3$ scales and $L = 4$ directions. There are totaly 125 scattering maps as sparse code candidates, namely $n_0 = 125$. For the convolution neural component, the number of neurons is set as $n_1 = 50$, and the size of filters is allocated as $m_1 = 9$ and $m_2 = 5$. We use this configuration to train three HWCNs corresponding to upscaling factor 2, 3, and 4, respectively.

Dataset We use the same training and testing sets as [5–7, 10, 11], where the training set consists of 91 natural images and the testing sets contain Set5 (5 images) and Set14 (14 images). Similarly, Set5 is used to evaluate the performance of upscaling factor 2, 3, and 4, and Set14 is used for factor 3.

Training procedure The training process is implemented on MATLAB with ScatNet [13] and MatConvNet [14] toolboxes. Similar to [5–7, 11], we concentrate on the illuminance channel and the two chrominance channels are bicubic interpolated for display. LR images x_i are obtained by first blurring original (HR) images y_i with the Gaussian kernel, then downsampling it by the upscaling factor, and upscaling it through the bicubic interpolation. We use Mean Squared Error (MSE) as loss function: $L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \|F_{x_i}(\theta) - y_i\|^2$, where F_{x_i} is the reconstructed HR image and $\theta = (W_1, B_1, W_2, B_2)$ denote all parameters for training. We employ the back propagation and stochastic gradient descent algorithm to minimize the loss function. To avoid the border effect, there is no padding for all the convolution layers. Fig. 4 shows the training convergence curve for upscaling factor 2. With the increment of number of iterations and training data, the proposed model will achieve better performance.

Experiment Results All the methods are implemented on MATLAB using the public source codes. Table 1 and Table 2 show results on testing set Set5 and Set14, respectively. The proposed method achieves the highest average PSNR for all scales. In Fig. 5, the proposed method achieves the best visual performance with sharpest edges and fewest artifacts, such as the eyebrow and the edge of hat.

5. CONCLUSION

This paper proposes a hierarchical end-to-end network structure, which comprises a scattering convolution part and a convolution neural part. The scattering convolution part is fulfilled by the pre-defined wavelet filters with nonlinear operators, and can be tailored in accordance with the frequency property to provide sparse code candidates. The convolution neural part could automatically select and weight these candidates for sparse coding. The proposed network could implement complex deep network with better generalization.

6. REFERENCES

[1] Stéphane Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.

[2] Joan Bruna and Stéphane Mallat, "Invariant scattering convolution networks," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1872–1886, 2013.

[3] David L Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] Emmanuel J Candès et al., "Compressive sampling," in *Proceedings of the international congress of mathematicians*. Madrid, Spain, 2006, vol. 3, pp. 1433–1452.

[5] Jianchao Yang, John Wright, Thomas Huang, and Y-i Ma, "Image super-resolution as sparse representation of raw image patches," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[6] Jianchao Yang, John Wright, Thomas S Huang, and Y-i Ma, "Image super-resolution via sparse representation," *Image Processing, IEEE Transactions on*, vol. 19, no. 11, pp. 2861–2873, 2010.

[7] Jianchao Yang, Zhaowen Wang, Zhe Lin, Scott Cohen, and Thomas Huang, "Coupled dictionary training for image super-resolution," *Image Processing, IEEE Transactions on*, vol. 21, no. 8, pp. 3467–3478, 2012.

[8] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*, pp. 711–730. Springer, 2012.

[9] Hong Chang, Dit-Yan Yeung, and Yimin Xiong, "Super-resolution through neighbor embedding," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, 2004, vol. 1, pp. I–I.

[10] R. Timofte, V. De, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 1920–1927.

[11] Chao Dong, Chen Change Loy, Kaiming He, and X-iaoou Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision–ECCV 2014*, pp. 184–199. Springer, 2014.

[12] Thomas Wiatowski and Helmut Bölcskei, "A mathematical theory of deep convolutional neural networks for feature extraction," *arXiv preprint arXiv:1512.06293*, 2015.

[13] J Andén, L Sifre, S Mallat, M Kapoko, V Lostanlen, and E Oyallon, "Scatnet," *Computer Software. Available: http://www. di. ens. fr/data/software/scatnet/.[Accessed: December 10, 2013]*, 2014.

[14] Andrea Vedaldi and Karel Lenc, "Matconvnet-convolutional neural networks for matlab," *arXiv preprint arXiv:1412.4564*, 2014.