ORTHOGONALLY REGULARIZED DEEP NETWORKS FOR IMAGE SUPER-RESOLUTION

Tiantong Guo, Hojjat S. Mousavi, Vishal Monga
The Pennsylvania State University

ABSTRACT

Deep learning methods, in particular trained Convolutional Neural Networks (CNNs) have recently been shown to produce compelling state-of-the-art results for single image Super-Resolution (SR). Invariably, a CNN is learned to map the low resolution (LR) image to its corresponding high resolution (HR) version in the spatial domain. Aiming for faster inference and more efficient solutions than solving the SR problem in the spatial domain, we propose a novel network structure for learning the SR mapping function in an image transform domain, specifically the Discrete Cosine Transform (DCT). As a first contribution, we show that DCT can be integrated into the network structure as a Convolutional DCT (CDCT) layer. We further extend the network to allow the CDCT layer to become *trainable (i.e. optimizable)*. Because this layer represents an image transform, we enforce pairwise orthogonality constraints on the individual basis functions/filters. This Orthogonally Regularized Deep SR network (ORDSR) simplifies the SR task by taking advantage of image transform domain while adapting the design of transform basis to the training image set. Experimental results show ORDSR achieves state-of-the-art SR image quality with fewer parameters than most of the deep CNN methods.

1. INTRODUCTION

Single Image Super-Resolution (SISR) has emerged as one of the most significant ill-posed imaging problems due to a variety of applications in civilian domains as well as in law enforcement [1]. With an increasing number of mobile cameras, generating a clean, sharp image with lower storage and computation requirements is highly desirable.

The single image SR task has been addressed by dictionary based and sparsity constrained learning methods and more recently via deep learning methods. A typical learning/example based SR approach employs two dictionaries of HR/LR images/patches [2, 3, 4, 5, 6]. These dictionaries are often learned with sparse coding methods to reconstruct the SR results. Many of these methods require handcrafted dictionary features which are not readily available [7].

Recently, deep learning methods have shown to produce compelling state-of-the-art SR results and across a variety of different image collections [8]. One of the earliest deep SR methods was SRCNN [9, 10] and its extensions that train multiple coupled networks have been pursued as well [11]. Other variants include [12, 13] which use self-similar patches to explore the self-example based SR idea. However, the network structures are no-less mutations of straightforward spatial mapping functions between LR/HR image. These spatial domain mappings were further boosted by global and local by-pass structures as introduced by residual learning [14]. Residual network structures essentially reduce the training burden (in the sense of learning complexity) of the deep CNN which is still constructed in spatial domain.

Motivation: Our work is motivated by the recent promising performance of SR methods in the transform domain [8]. Our goal is faster inference and structures with fewer parameters than existing spatial domain CNNs. Specifically, the Discrete Wavelet Transformation (DWT) has been explored for the SR problem in traditional frameworks [15, 16, 17, 18] and more recently also in deep networks [19].

In this paper, we begin by exploring a DCT domain deep SR method. In the DCT domain, the differences between a given LR-HR image pair is the missing high-frequency information while they typically share the same low-frequency signature (see analysis in section 2). Because the low-to-high-resolution mapping is simpler, the learning burden of the network can be reduced and both the convergence rate and inference of the network can become faster. As a first contribution, we show that DCT can be integrated into the network structure as a convolutional DCT (CDCT) layer. We further extend the network to allow the CDCT layer to become *trainable (i.e. optimizable)*. Because this layer represents an image transform, we enforce pairwise orthogonality constraints on the individual basis functions/filters. This Orthogonally Regularized Deep SR network (ORDSR) simplifies the SR task by taking advantage of image transform domain while adapting the design of transform basis to the training image set.

The main contributions of this paper are as follows:

1. We propose a novel network structure that attacks SR problem in the image transform domain;
2. We build a special CDCT layer integrating DCT procedure into the network, where the CDCT filters are adaptable and trainable;
3. We add novel orthogonality constraints on the newly introduced 'transform layer' to maintain the pairwise orthogonality properties of the learned basis.

To the best of our knowledge, ORDSR network is the first approach that allows optimization of basis functions for transform domain image SR within a deep learning framework.

This work is supported by an NSF CAREER Award to V. Monga.

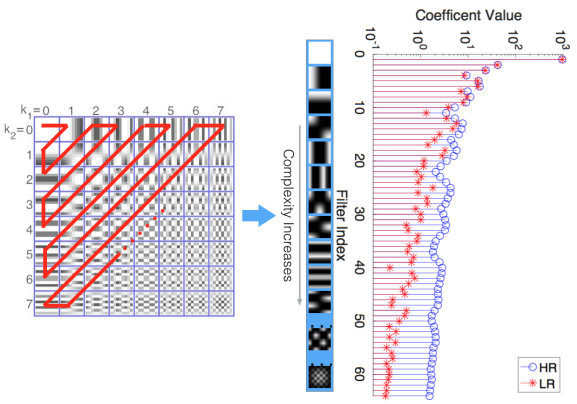


Fig. 1: *Left*: zig-zag reorder of the DCT basis family. *Right*: average coefficient values generated by $\{w_i\}_{i=1}^{64}$ of *lenna.bmp*.

2. SUPER-RESOLUTION IN DCT DOMAIN

An image $x(n_1, n_2)$ of size $H \times W$ can be decomposed into $H/N \times W/N$ blocks of size $N \times N$. For the $(m, n)^{th}$ block, the DCT coefficients are computed as:

$$X_{m,n}(k_1, k_2) = \sum_{n_2=0}^{N-1} \sum_{n_1=0}^{N-1} x_{m,n}(n_1, n_2) \times w_{k_1,k_2}(n_1, n_2) \quad (1)$$

where $k_1, k_2, n_1, n_2 = 0, \ldots, N-1$, and $w_{k_1,k_2}(n_1, n_2)$ is the DCT basis function, specifically DCT-II basis, defined as:

$$w_{k_1,k_2}(n_1, n_2) = C_{k_1,k_2} cos\left[\frac{\pi}{N}\left(n_1 + \frac{1}{2}\right)k_1\right] cos\left[\frac{\pi}{N}\left(n_2 + \frac{1}{2}\right)k_2\right] \quad (2)$$

where $C_{k_1,k_2} = \frac{\sqrt{1+\delta_{k_1}}\sqrt{1+\delta_{k_2}}}{N}$ and $\delta_k = 1$ if $k = 0$, $\delta_k = 0$ otherwise. For $N = 8$, there are 8×8 DCT bases and each basis w_{k_1,k_2} is of size 8×8.

Basis functions $\{w_{k_1,k_2}\}_{k_1,k_2=1,1}^{N,N} \in \mathbb{R}^{N \times N}$ are pairwise orthogonal, forming an orthogonal basis family:

$$< w_{k_1,k_2}, w_{l_1,l_2} >= \begin{cases} 1, & \text{if } k_1 = l_1, \text{ and } k_2 = l_2 \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

Corresponding to the DCT, the inverse DCT (IDCT) for the $(m, n)^{th}$ block is computed as:

$$x_{m,n}(n_1, n_2) = \sum_{k_2=0}^{N-1} \sum_{k_1=0}^{N-1} X_{m,n}(k_1, k_2) \times w_{k_1,k_2}(n_1, n_2) \quad (4)$$

Note that classical DCT is performed on $N \times N$ blocks of the original image. We now develop a reorganization of the DCT coefficients and their computation, which we show in Section 3 helps facilitate the implementation of DCT within a CNN.

Zig-zag reorder: We treat DCT basis functions as filters and reorganize them in a zig-zag order as shown in Fig. 1. The zig-zag function maps $\{w_{k_1,k_2}\}_{k_1,k_2=1,1}^{N,N}$ to $\{w_i\}_{i=1}^{N \times N}$. Specially, with the zig-zag mapping, as the index i increases, the complexity of w_i also increases, *i.e.* the lower end of $\{w_i\}_{i=1}^{64}$ is corresponding to low-frequency filters, while the higher end (bigger i) represents the high-frequency ones.

Given an HR image y, and its LR version x, we can plot the average coefficient values generated by the DCT filters $\{w_i\}_{i=1}^{N \times N}$, as shown in Fig. 1. As the plot suggests, the HR image y and the LR image x share the same low-frequency spectra, while x has less high-frequency information than y. With the help of DCT filters, SR becomes a problem of recovering high-frequency DCT coefficients of the HR image from the corresponding ones of the LR input.

3. CONVOLUTIONAL DCT LAYER WITH THE ORTHOGONALITY CONSTRAINTS

To integrate the DCT analysis within a CNN, we construct a convolutional DCT (CDCT) layer.

Initialization: The CDCT layer is initialed using the DCT basis $\{w_i\}_{i=1}^{N \times N}$. For $N = 8$, there are 64 filters $\{w_i\}_{i=1}^{64}$ of size 8×8 in the CDCT layer such that the complexity (high-frequency content) increases with the filter index.

Unlike classical DCT that produces 8×8 block-wise DCT coefficients, the CDCT layer produces 64 frequency maps $\{f_i\}_{i=1}^{64}$ for the whole image by convolving $\{w_i\}_{i=1}^{64}$ with the input image x as shown in Eq. (5).

$$f_i = w_i * x, \forall i \in [1, ..., 64] \quad (5)$$

These maps, $\{f_i\}_{i=1}^{64}$, form a cube called DCT cube. The DCT cube is essentially a reorganized version of classical block-wise DCT coefficients of the whole image.

As i increases, f_i corresponds to higher frequency components of the whole image. Thus, we divide the DCT cube into two parts by a threshold T, namely low-frequency spectral maps $f_{low} = \{f_i\}_{i=1}^{T}$ and high-frequency spectral maps $f_{high} = \{f_i\}_{i=T+1}^{64}$.

The CDCT layer can also perform IDCT by transpose convolving[1] $\{w_i\}_{i=1}^{64}$ with the DCT cube $\{f_i\}_{i=1}^{64}$ respectively, resulting in the spatial image y. This procedure can be viewed as a convolution of w_i with a 8-zero padded f_i:

$$y = \sum_{i=1}^{64} w_i * g(f_i) \quad (6)$$

where $g(\cdot)$ is the zero padding function. For details on the implementation of both the DCT and IDCT as a CNN layer we refer the reader to our accompanying technical report [22].

Orthogonality Constraints: The aforementioned CDCT layer can in fact be learned. Consistent with classical DCT, we enable learning but in the presence of pairwise orthogonality constraints. These constraints are captured by a regularization term which is added to the network's total cost function – see Eq. (8). As suggested in (3), any distinct filter pairs in the CDCT layer should have a zero inner product. Here, the inner product is computed by vectorized multiplication between two filters. Ideally, ϵ should be zero but may be relaxed slightly in practice for numerical optimization.

$$\forall i \neq j, \|vec(w_i)^T vec(w_j) - \epsilon\|_2^2 = 0 \quad (7)$$

[1] Some literature [20, 21] refer this procedure as deconvolution, fractionally stride convolution or backward convolution in neural network setups.

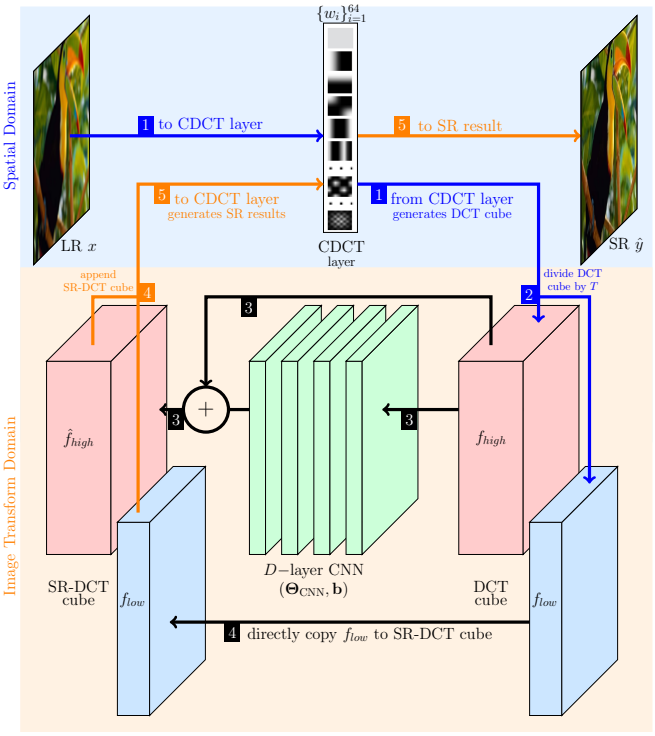


Fig. 2: The ORDSR network structure. Please refer to color version to follow the network flow. The CDCT layer serves two purposes: producing DCT cube (following blue arrow) and generating SR image from a SR-DCT cube (following orange arrow).

4. ORDSR NETWORK STRUCTURE

The ORDSR network has two parts: a CDCT layer and a $D-$layer CNN. The CDCT layer produces both the DCT cube of the input image and generates the SR results from the CNN's output DCT cube. The CNN recovers the high-frequency spectra by generating a SR-DCT cube.

Fig. 2 shows the structure of the ORDSR network with $N = 8$. For an input LR image x, the goal of ORDSR is to generate its SR version $\hat{y}$ as follows:

1. The input LR image x is convolved with CDCT layer producing DCT cube $\{f_i\}_{i=1}^{64}$ as shown in (5);
2. The DCT cube of x is divided as f_{low} and f_{high} corresponding to low and high-frequency spectra by an index threshold T, based on description in Section 3;
3. The CNN takes f_{high} as input and recovers the missing high-frequency information using a residual network structure, generating $\hat{f}_{high}$;
4. The $\hat{f}_{high}$ is appended by f_{low} forming the SR-DCT cube $\{\hat{f}_i\}_{i=1}^{64}$. As the f_{low} is unchanged between x and its corresponding HR image y, only f_{high} needs to be modified for generating $\hat{y}$;
5. The SR-DCT cube $\{\hat{f}_i\}_{i=1}^{64}$ is transpose convolved with CDCT layer (to perform the IDCT) generating $\hat{y}$, as shown in (6).

In step 3, only taking the f_{high} components of the DCT cube reduces the input channel numbers for the CNN, which makes the training procedure faster. In step 4, the CNN uses a residual network structure to further reduce the computational burden. Steps 1 and 5 are performed in the image spatial domain while steps 2-4 are in the image transform domain.

The inference procedure is denoted as $h_{(\mathbf{\Theta},\mathbf{b})}(x) = \hat{y}$, where $(\mathbf{\Theta}, \mathbf{b})$ is the collection of all the trainable filter weights and biases of ORDSR network. Note that $\mathbf{\Theta} = \{\mathbf{\Theta}_{\text{CNN}}, \{w_i\}_{i=1}^{64}\}$ as shown in Fig. 2.

We develop a modified back-propagation scheme [22] which enables the proposed ORDSR to minimize:

$$\begin{aligned}\mathbf{\Theta}, \mathbf{b} = \underset{\mathbf{\Theta},\mathbf{b}}{argmin} \frac{1}{2}\|h_{(\mathbf{\Theta},\mathbf{b})}(x) - y\|_2^2 + \sigma \sum_t \|\Theta_t\|_2^2 \\ + \gamma \sum_{(i,j)} \|vec(w_i)^T vec(w_j) - \epsilon\|_2^2\end{aligned} \tag{8}$$

where $\{(i,j)\}$ are all the unique pairwise indexes of the $\{w_i\}$ and t is the collective index of all trainable filter weights in $\mathbf{\Theta}$. ORDSR also utilizes an ℓ_2 regularization of weights with a trade off parameter σ. Note $w_i \in \mathbf{\Theta}$, thus filters of CDCT layer are updated to generate a better $\hat{y}$.

5. EXPERIMENTAL RESULTS

Data preparation: The 291 images dataset [23] is used for training. The images are augmented by rotating the images by $\{90°, 180°, 270°\}$ and scaling by factors of $\{0.7, 0.8, 0.9\}$. The augmented images are down-sampled and subsequently enlarged using bicubic interpolation to form the LR training images. All the LR/HR images are further cropped into 40×40 pixels sub-images with 10 pixels overlap for training. During the test phase, Set5 [24] and Set14 [25] are used to evaluate our proposed method. Both training and testing phases of ORDSR only utilize the luminance channel information. For color images, Cb, Cr channels are enlarged by bicubic interpolation.

Training Settings: During the training process, the gradients are clipped to 0.01 and the Adam optimizer [26] is adopted to update $(\mathbf{\Theta}, \mathbf{b})$. The initial learning rate is 0.001 and decreases by 25% every 25 epochs. σ is set to 1×10^{-3} to prevent over-fitting. The CNN has $D = 14$ same-sized convolutional hidden layers with filter size of $3 \times 3 \times 64$. This configuration results in a network with only 75% of the parameters in the state-of-the-art method VDSR [14]. The ORDSR is implemented with TensorFlow [27] packages on one TITAN X GPU for both the training and testing, which takes 5 hours to reach 85 epochs for the reported results.

SR Results:[2] Table 1 shows the comparison of ORDSR with other state-of-the-art methods: classical methods ScSR [5] and A+ [28], deep learning based methods SelfEx [12], FSRCNN [10], SRCNN [9] and VDSR. The metrics used for image quality assessment are PSNR and SSIM [29]. The comparison is constrained among methods with the same training set and same computational requirements. ORDSR

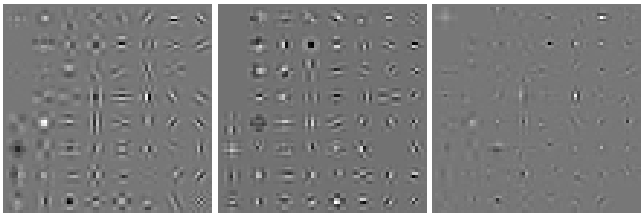
[2]Code available on `http://signal.ee.psu.edu/ORDSR.html`

Table 1: PSNR and SSIM comparisons. (The best results are shown in bold **red** and the second best are shown in blue.)

PSNR | SSIM	Scale	Bicubic [Baseline]		ScSR [TIP 10]		A+ [ACCV 14]		SelfEx [CVPR 15]		FSRCNN [ECCV 16]		SRCNN [PAMI 16]		VDSR [CVPR 16]		ORDSR [Proposed]	
Set5	x2	33.64	0.9292	35.78	0.9485	36.55	0.9544	36.47	0.9538	36.94	0.9558	36.66	0.9542	37.52	**0.9586**	**37.53**	0.9574
	x3	30.39	0.8678	31.34	0.8869	32.58	0.9088	32.57	0.9092	33.06	0.9140	32.75	0.9090	33.66	0.9212	**33.74**	**0.9221**
	x4	28.42	0.8101	29.07	0.8263	30.27	0.8605	30.32	0.8640	30.55	0.8657	30.48	0.8628	31.35	0.8820	**31.45**	**0.8847**
Set14	x2	30.22	0.8683	31.64	0.8940	32.29	0.9055	32.24	0.9032	32.54	0.9088	32.42	0.9063	33.02	0.9102	**33.04**	**0.9109**
	x3	27.53	0.7737	28.19	0.7977	29.13	0.8188	29.16	0.8196	29.37	0.8242	29.28	0.8209	29.75	0.8294	**29.81**	**0.8300**
	x4	25.99	0.7023	26.40	0.7218	27.33	0.7489	27.40	0.7518	27.50	0.7535	27.40	0.7503	28.01	0.7662	**28.06**	**0.7664**

Table 2: Average results on Set14 with scale factor 3

	$\gamma = 1$						$\gamma = 0$	$w_i \notin \mathbf{\Theta}$
$\epsilon =$	0.0001	**0.001**	0.01	0.1	0.5	1		
PSNR	29.7932	**29.8104**	29.7815	29.7805	29.7786	29.7208	29.7621	29.7165
SSIM	0.8295	**0.8300**	0.8281	0.8265	0.8266	0.8252	0.8201	0.8189



(a) $\epsilon = 0.0001$ (b) $\epsilon = 0.001$ (c) $\epsilon = 0.01$

Fig. 3: Learned filters of CDCT layers with different ϵ. (Filters are normalized and reordered for display purpose.)

produces best results using only 75% the number of the parameters than VDSR. Fig. 5 displays the testing image in detail, ORDSR generates more defined edges with better quality assessments among the competing methods.

CDCT Layer: With different ϵ, the orthogonality constants have different effects on the CDCT layer. As shown in Table 2, with $\gamma = 1$, a very small or very big ϵ will end up with a tightly constrained or relaxed CDCT layer. Fig. 3 shows smaller ϵ preserves more DCT filters structure within the CDCT layer. Cross validation shows $\epsilon = 0.001$ produces the best results. With $\gamma = 0$, the ORDSR is trained without the orthogonality constraints, which produced less favorable results showing the importance of CDCT layer being orthogonal. Also if we excludes w_i from $\mathbf{\Theta}$, the ORDSR is trained without updating the CDCT layer at all. Table 2 shows the importance of CDCT layer being adaptively trainable.

Threshold T: Fig. 4 shows the effects of T over the PSNR of the SR results. A smaller T implies a smaller fraction of f_{low} is directly copied to SR-DCT cube as described in the inference step 3. However, after $T < 4$, decreasing the threshold does not change the SR image quality significantly. This further shows that the low frequency spectra between LR/HR image are indeed shared. All reported results use $T = 4$.

6. CONCLUSION

We propose a novel network structure to tackle SR problem in the image transform domain. We show that DCT can be integrated into the network structure as a Convolutional DCT (CDCT) layer. We further extend the network to allow the CDCT layer to become *trainable (i.e. optimizable)*. Experimental results show the effectiveness of performing SR in the image transform domain by ORDSR, also the significance of ORDSR learning bases that are specific for natural image SR.

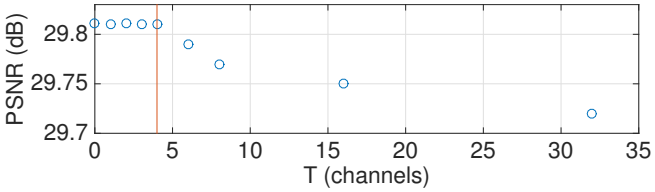


Fig. 4: Avg. PSNR of Set14 with scale factor 3 on different T. When $T < 4$, decreasing T will not affect the SR results.

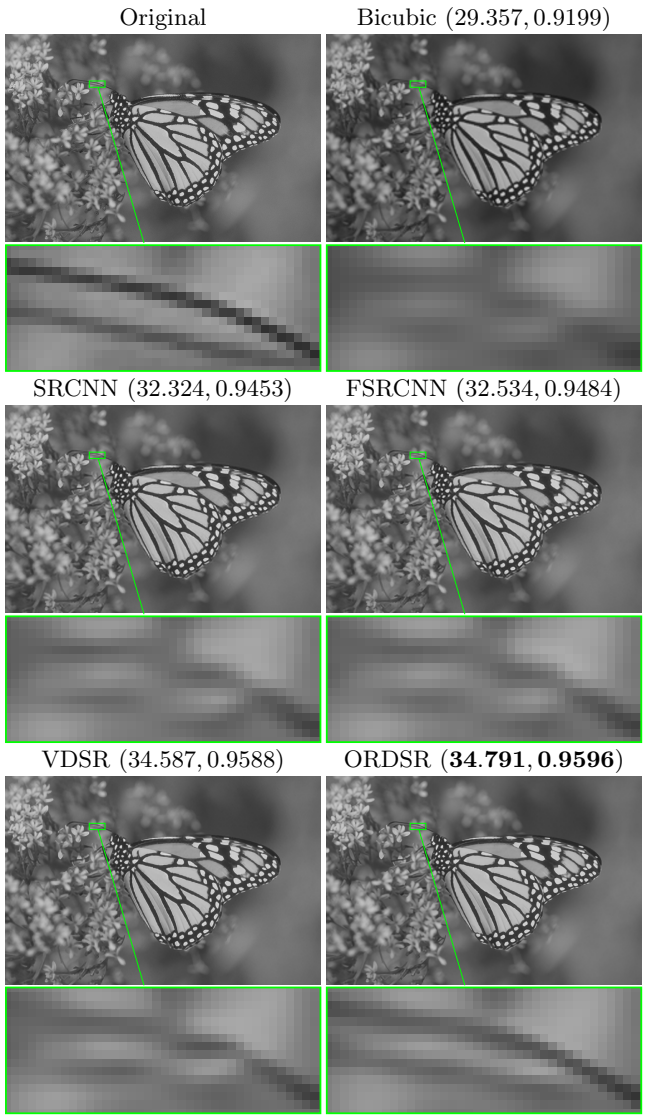


Fig. 5: The SR results of test image *monarch.bmp* of scale factor 3. The image metrics is shown as (PSNR, SSIM). ORDSR produces best visual results with better quality assessments.

7. REFERENCES

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *Signal Processing Magazine, IEEE*, vol. 20, no. 3, pp. 21–36, 2003.

[2] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators," *Image Processing, IEEE Transactions on*, vol. 19, no. 11, pp. 2889–2900, 2010.

[3] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2004, vol. 1, pp. I–I.

[4] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Computer Vision, IEEE International Conference on*, 2009, pp. 349–356.

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

[6] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 6, pp. 1127–1133, 2010.

[7] L. Zhang and W. Zuo, "Image restoration: From sparse and low-rank priors to deep priors, lecture notes," *Signal Processing Magazine, IEEE*, vol. 34, no. 5, pp. 172–179, 2017.

[8] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, et al., "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Computer Vision and Pattern Recognition Workshops, IEEE Conference on*, July 2017.

[9] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision, ECCV*, pp. 184–199. Springer, 2014.

[10] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Computer Vision, ECCV*, pp. 391–407. 2016.

[11] T. Guo, H. S. Mousavi, and V. Monga, "Deep learning based image super-resolution with coupled backpropagation," in *Signal and Information Processing, IEEE Global Conference on*, 2016, pp. 237–241.

[12] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2015, pp. 5197–5206.

[13] Z. Wang, Y. Yang, Z. Wang, S. Chang, W. Han, J. Yang, and T. S. Huang, "Self-tuned deep super resolution," *arXiv preprint arXiv:1504.05632*, 2015.

[14] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2016, pp. 1646–1654.

[15] S. Zhao, H. Han, and S. Peng, "Wavelet-domain hmt-based image super-resolution," in *Image Processing, IEEE International Conference on*, 2003, pp. II–953.

[16] M. D. Robinson, C. A. Toth, J. Y. Lo, and S. Farsiu, "Efficient fourier-wavelet super-resolution," *Image Processing, IEEE Transactions on*, vol. 19, no. 10, pp. 2669–2681, 2010.

[17] M. E.-S. Wahed, "Image enhancement using second generation wavelet super resolution," *International Journal of Physical Sciences*, vol. 2, no. 6, pp. 149–158, 2007.

[18] H. Ji and C. Fermüller, "Robust wavelet-based super-resolution reconstruction: theory and algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 4, pp. 649–660, 2009.

[19] T. Guo, H. S. Mousavi, T. H. Vu, and V. Monga, "Deep wavelet prediction for image super-resolution," in *Computer Vision and Pattern Recognition Workshops, IEEE Conference on*, 2017, pp. 1100–1109.

[20] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Computer Vision, IEEE International Conference on*, 2015, pp. 1520–1528.

[21] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[22] T. Guo, H. S. Mousavi, and V. Monga, "A technical report on: orthogonally regularized deep networks for image super-resolution," 2017, Code available on http://signal.ee.psu.edu/ORDSR.html.

[23] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2015, pp. 3791–3799.

[24] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[25] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] M. Abadi, A. Agarwal, and P. B. et. al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[28] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Computer Vision, ACCV*, pp. 111–126. Springer, 2014.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.