CGI2018 manuscript No.
(will be inserted by the editor)

Efficient Image Super Resolution Integration

Ke Xu[1,3] · **Xin Wang**[1] · **Xin Yang**[1*] · **Shengfeng He**[2*] · **Qiang Zhang**[1] · **Baocai Yin**[1] · **Xiaopeng Wei**[1] · **Rynson W.H. Lau**[3]

Abstract The Super Resolution (SR) problem is challenging due to the diversity of image types with little shared properties as well as the speed required by online applications, e.g., target identification. In this paper, we explore the merits and demerits of recent deep learning based and conventional patch-based SR methods, and show that they can be integrated in a complementary manner, while balancing the reconstruction quality and time cost. Motivated by this, we further propose an integration framework to take the results from FSRCNN and A+ methods as inputs, and directly learn a pixel-wise mapping between the inputs and the reconstructed results using the Gaussian Conditional Random Fields (GCRFs). The learned pixel-wise integration mapping is flexible to accommodate different upscaling factors. Experimental results show that the proposed framework can achieve superior SR performance compared with the state-of-the-arts while being efficient.

Keywords Image super resolution · image processing · Gaussian conditional random fields.

Ke Xu, Xin Wang, Xin Yang, Qiang Zhang, Baocai Yin and Xiaopeng Wei
E-mail: kkangwing@mail.dlut.edu.cn, wangxinlp@gmail.com, xinyang@dlut.edu.cn, zhangq@dlut.edu.cn, ybc@dlut.edu.cn and weixp@dlut.edu.cn
Shengfeng He
E-mail: hesfe@scut.edu.cn
Rynson W.H. Lau
E-mail: Rynson.Lau@cityu.edu.hk
[1] Dalian University of Technology, China.
[2] South China University of Technology, China.
[3] City University of Hong Kong, Hong Kong.
* Xin Yang and Shengfeng He are the corresponding authors.

1 Introduction

Super resolution (SR) is to reconstruct high resolution (HR) images from low resolution (LR) inputs. It is a challenging yet important research problem, with lots of applications ranging from medical to social security. Various methods have been proposed to improve the reconstruction quality and speed up the reconstruction time.

Conventional patch-based SR works [3, 31, 37, 43] have shown that, with the basic assumption that images of complicated contents can be represented by the combination of finite basic local structures, we can solve the SR problem via learning the relationship between the input LR patches and the ground truth image patches, and enjoy the benefit of effectively finding good local image priors as the basis. One main shortage of these methods is that they rely on additional strong assumptions, e.g., local self similarity [16], and their performance decrease if such assumptions are violated. These methods are also limited by their abilities of modeling non-linear mapping relations.

Recent success in deep learning technology helps further improve the SR reconstruction quality. A deep model consists of multi-layers, with each layer learning the abstract representations of data. They have been shown to achieve great success in various disciplines, including speech recognition [2], object identification [42], and portrait matting [32]. They have also been used for image super resolution in the pioneering work by Dong *et al.* [5, 6], termed as Super Resolution Convolutional Neural Network (SRCNN). SRCNN has drawn much attention due to its simple architecture but superior reconstruction quality. Some variant networks of SRCNN have been proposed, including [7, 21, 22, 34, 40], to further improve the SR performance. However, as the net-

work goes deeper, the time consumption may become a more important issue. For example, Tai *et al.* [34] propose a very deep neural network (up to 52 convolutional layers) to learn the residuals between LR/HR images. Although recursive learning is applied to control their model parameters, their very deep network structure results in a severe computation overhead. It requires tens of seconds CPU time to process one image, which is not acceptable for most online applications. It is worth noting that although deep learning based methods can be efficient when running on GPUs, CPU test time is a very important issue in practical SR applications where GPUs are not available.

Motivations. Although these two types of SR methods have their own limitations, we observe that the faithful local image priors of patch-based methods and the global non-linear LR/HR mapping ability of deep learning based methods may complement each other. In order to better integrate the advantages of these two kinds of methods, we investigate A+ [38], a patch-based methods, and FSRCNN [7], a deep learning method.

A+. It assumes that all LR/HR patches lie on the manifold in the LR/HR space, so that input patches can be approximated by existing database patches on the manifold. HR outputs of A+ are thus reconstructed by the existing database patches retrieved, which can provide reliable basic priors of local image contents, e.g., local image contents along sharp boundaries are reconstructed with less visual artifacts. On the contrary, RFL [31] do not follow the manifold assumption and directly learn the LR/HR patch mappings using random forests, and we can also observe local artifacts (e.g., Figures 1(d) and 1(i)).

FSRCNN. In order to reduce the time cost during the SR process, FSRCNN [7] confines its model scale in the specific hourglass-shape network structure. Its ability of non-linear mapping modeling is restricted accordingly. Although it provides an overall higher SR performance than A+ [38], it behaves as low-pass filters when processing image structures with dramatic transitions, where we observe ringing artifacts. Similar phenomena are also observed in SRCNN (e.g., Figures 1(g) and 1(h)).

In this paper, we propose an integration framework to address the single image super resolution problem. Our integration framework facilitates the exploration of various integration strategies of the state-of-the-art SR methods, and learns how these integrated methods interact with each other to produce the reconstructed results. Extensive experiments show that the proposed framework achieves good performance (quantitatively and qualitatively) compared with the state-of-the-arts, while being efficient.

The main contributions of this paper are summarized as:

- We demonstrate the complementarity between the patch-based and the deep learning-based SR methods from the global and local perspectives.
- We propose a GCRF-based integration model that learns the probabilistic distribution of the interaction between FSRCNN and A+, with a minimum time cost.
- We investigate different integration methods and explore the importance of conventional SR algorithms to the reconstruction quality.

2 Related Work

Given an input LR image, the SR process attempts to recover an HR image as output. There are many approaches to the SR problem. This section summarizes the main works. Readers may refer to [10,27] for a comprehensive study of the SR problem.

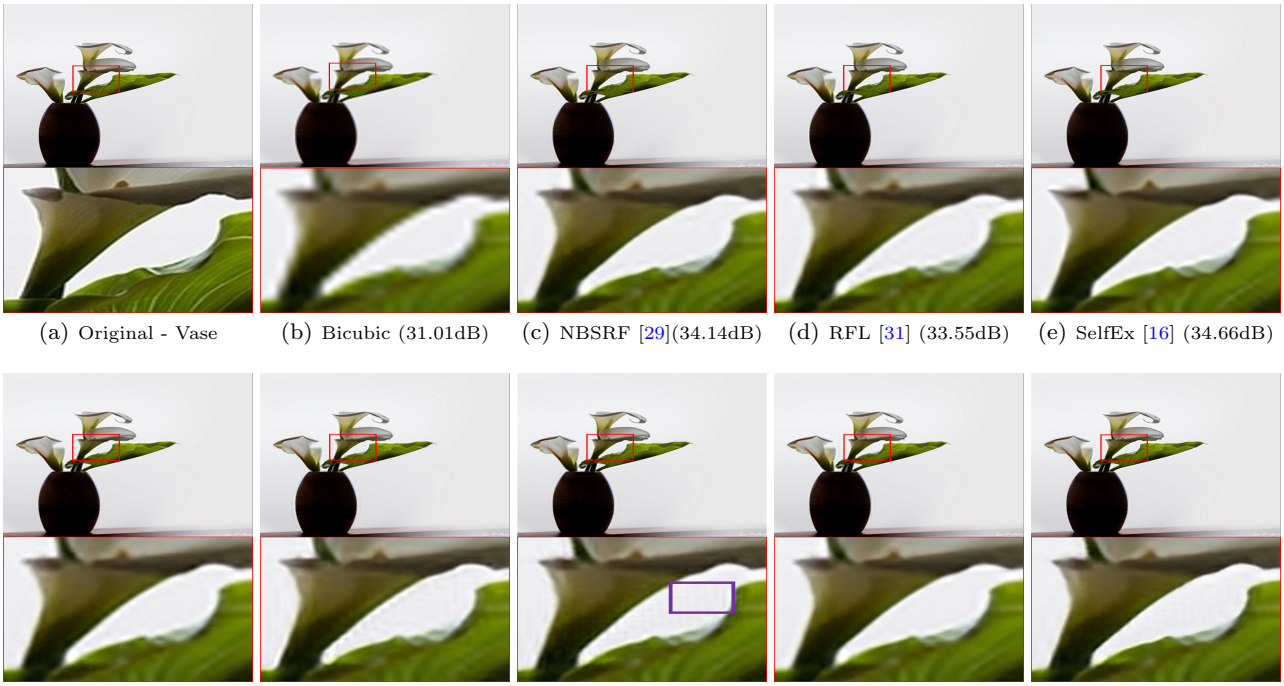
2.1 Conventional Methods

Earlier SR methods are mainly based on image interpolation and sharpening, including nearest-neighbor, bilinear, bicubic and Lanczos [8,36]. They are widely used because of their efficiency as well as their advantages on recovering the smooth regions. However, as they assume that images are smooth or band-limited, the resulting HR images inevitably exhibit visual artifacts, such as blurring, ringing, blocking and aliasing. One solution to such artifacts is to introduce strong prior information, e.g., edge prior [4] and edge statistics [11]. Another solution is to apply blur kernel estimation [9, 26], such that the estimated blur kernels can be used to improve the blurring artifacts.

2.2 Learning Based Methods

Recent SR methods mainly focus on recovering the LR/HR mapping relations. In general, there are two ways to derive these mapping relations: external and internal databases.

A large number of methods have been proposed to learn the image priors from external databases. They may vary from using different complex models to using different LR/HR patch encoding strategies. For example, Kim *et al.* [23] and Wang *et al.* [39] use Kernel Principal Component Analysis to obtain the prior knowledge. Yang *et al.* [43] seek sparse representations of LR/HR patch pairs (dictionary-pairs) to learn the



(a) Original - Vase (b) Bicubic (31.01dB) (c) NBSRF [29](34.14dB) (d) RFL [31] (33.55dB) (e) SelfEx [16] (34.66dB)

(f) JOR [3] (33.55dB) (g) SRCNN [6] (33.62dB) (h) FSRCNN [7](34.23dB) (i) A+ [38] (33.93dB) (j) Ours (34.95dB)

Fig. 1: The proposed framework clearly improves the reconstruction quality of the state-of-the-art SR methods. Even when FSRCNN [7] or A+ [38] performs worse than SelfEx [16], our framework can still exploit their advantages to perform better. Images are 4× upscaled.

mapping relation. Freeman *et al.* [13] obtain a pool of LR/HR training patches to infer output HR patches which corresponding LR patches are the nearest neighbors to the input LR patches. As local patch information alone is insufficient for the SR process, they apply a Markov network to take into account spatial neighborhood effects. Based on the Markov network, Freeman *et al.* [14] further extract different components to encode the LR/HR patches. Chang *et al.* [1] extract the k-nearest neighbors (kNN) for an input LR patch and embed them into the local coordinate system using Locally Linear Embedding (LLE) [28]. The output HR patch is then reconstructed from those coordinates.

Regression approaches are also very popular for solving the SR problem, e.g., locally linear regression [37], adjusted anchored neighborhood regression [38], joint regression [3], Kernel Ridge Regression [24] and random forests [31].

In general, the SR methods using external databases can achieve good performance as a large diversity of prior knowledge can be obtained from the external databases. However, these algorithms need to search huge databases for matching HR patches during the SR process, which is inefficient. In addition, they typically require re-training of the learned models whenever a new scaling factor is used. Nevertheless, as these HR/LR encoding strategies are based on the assumptions that some components are more descriptive than the others in encoding the HR/LR patches for certain types of images, these assumptions can be combined together to provide more comprehensive information for the SR process. We thus want to exploit these existing methods to combine their advantages.

Learning methods based on internal databases try to learn the priors directly from the input images, by taking advantage of the recurrence of patches within the input images to construct the database. For example, Freedman *et al.* [12] exploit the local self similarity within the LR image through a pyramid of downscaled images. Michaeli *et al.* [26] treat the SR process as a blur problem and try to recover the PSF kernel by learning the mapping relation. Huang *et al.* [16] apply a modified PatchMatch algorithm to warp each LR patch to find its best matching patch in the input LR image and then unwarp the matching patch as the HR patch. All these methods avoid the high cost in computing the image priors, but the quality of their reconstruction results inevitably deteriorates if the input images contain sparse self-similarity.

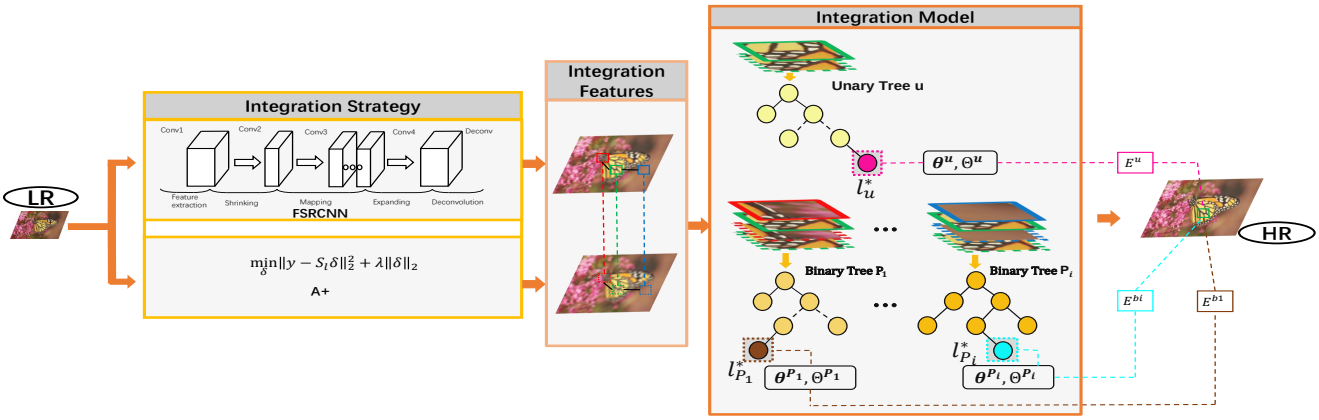


Fig. 2: Overview of the proposed integration framework. The input LR image X_L is first fed to the FSRCNN network to obtain feature-image X_F and concurrently to the A+ algorithm to obtain feature-image X_A, both of the same size as Y. During the integration, each pixel in X_A together with its corresponding pixel in X_F are used to predict the pixel in the final HR image Y from a trained pixel-wise integration model. Specifically, the cascaded pixels are used to traverse both the unary and binary trees to obtain the local energy terms, which are then globally optimized to reconstruct the final HR images.

2.3 Deep Learning Based Method

Deep learning techniques have achieved great success on the SR task, with the pioneer work being SRCNN [5,6]. Unlike the conventional SR methods, SRCNN directly learns the non-linear Bicubic-upscaled LR/HR mapping relations. This end-to-end learning ensures an accurate inference. Wang *et al.* [40] propose a sparse coding based network (SCN) to further exploit the inherent relationship between SRCNN and the sparse-coding domain expertise. Deeper network structures have also been explored in [21, 22]. In particular, both DRCN [21] and DRRN [34] expand SRCNN to a very deep structure by recurrently using the convolutional layers in different manners. They achieve high reconstruction quality but at much higher computational costs. Dong *et al.* [7] propose an accelerated fast network, called FSRCNN. Unlike the former deep learning based SR methods, FSRCNN directly performs on the original LR images. In addition, it also proposes an hourglass-shape network structure, which achieves the state-of-the-art reconstruction accuracy at a lower time cost.

2.4 Multi-Frame Based Methods

In this paper, we focus on solving the SR problem by combining Single Image Super Resolution (SISR) methods. The SISR methods generally take one input LR image and output a single HR image, while multi-frame SR methods [19,20] take a sequence of frames as inputs and exploit the complementarity among them. In contrast, we investigate in this work the complementarity among different SISR methods.

3 Methodology

3.1 Overview

In this paper, we propose an integration framework to estimate the pixel values for the output HR image. Our work is in the YCbCr color space, as humans are more sensitive to luminance changes [31]. Our framework has three components, *integration strategy*, *integration features* and *integration model*, as shown in Figure 2.

Our framework combines the advantages of the FSRCNN and A+ methods. First, we apply A+ [38] and FSRCNN [7] on the input LR image individually to increase its resolution to the expected size. We refer to these images as integration feature-images. Second, for each pixel, we construct a feature vector from the corresponding pixel values of the feature-images. This pixel-based feature vector consists of the unary feature and the binary features with its neighborhood. As shown in Section 4, this form of feature integration can lead to efficient learning as well as testing processes. Finally, we learn the mapping from these feature vectors to the reconstructed pixel values.

To capture the regression relation between the feature vectors and the pixel values in the output HR image, we construct an integration model based on the Gaussian Conditional Random Field. As demonstrated in Figure 1 (purple box), while FSRCNN predicts well

at shape edges (Figure 1(h)), it suffers from ringing artifacts in the shift regions between object edges and background. In contrast, A+ generates much smoother results at those regions (Figure 1(i)), but is less satisfactory at high frequency details.

3.2 Integration Model

Our integration model is based on the Gaussian Conditional Random Field (GCRF) [35], which is one of the most successful Probabilistic Graphical Models and good at describing the relations of both individual and pairwise variables. In our case, natural images are intrinsically structural and pixels are highly related to their neighborhoods. Hence, estimating an integrated pixel should not consider only the features of the pixel itself but also the features of its neighborhoods. We find this model well suited for our goal.

Specifically, we consider the entire reconstructed image Y as a graph and each pixel $Y(p)$ as a node. Each node is assigned with a set of vectorized feature-pixels. Our task is then to infer the pixel values of the reconstructed HR image, given the combined feature-pixels. Given the integration-images X_1, ..., X_n, where n=2 if we only integrate A+ and FSRCNN, the joint conditional probability of the combined values in Y is defined as:

$$P(Y|\{X_i\}_{i=1:n}, W) \propto \exp\{-E(Y|\{X_i\}_{i=1:n}, W)\}, \quad (1)$$

where W refers to the model parameters. The energy function is composed of a collection of local energy terms at each node (i.e., at each pixel) and at each pair of neighboring nodes (i.e., at each pair of neighboring pixels) as:

$$E(Y|\{X_i\}_{i=1:n}, W) = \sum_{p\in Y} E^u(Y(p)|\{X_i(p)\}_{i=1:n}, W_u) + \sum_{p\in Y q\in N(p)} E^b(Y(p), Y(q)|\{X_i(p), X_i(q)\}_{i=1:n}, W_b), \quad (2)$$

where $N(p)$ denotes the set of neighboring nodes for each node p. W_u and W_b refer to the unary and binary parameters, respectively, in our model. E^u and E^b refer to the local energy terms for the individual and pairwise relationship.

Unary Term E^u. To model how the pixel values at each node in the final estimation depends on the pixel value from each individual feature-image, we define the local unary energy function E^u based on the concatenated unary feature vector $f(p)$ in Eq. 2 as a local Gaussian model:

$$E^u(Y(p)|\{X_i\}_{i=1:n}; W_u) \stackrel{def}{=} \Theta_p^u(Y(p) - \theta_p^u f(p))^2, \quad (3)$$

where Θ_p^u and $\theta_p^u f(p)$ refer to the inverse of variance and the mean of the local Gaussian model (i.e., parameters W_u).

Here, the local unary energy function functions as a local classifier, similar to [15, 41], which takes a node $X(p)$ and its integration-feature vector $f(p)$ as input and computes a probability distribution of the pixel value for that node as:

$$P(Y(p)|\{X_i\}_{i=1:n}; W_u) = \exp\{-E^u(y_i|\{X_i\}_{i=1:n}; W_u)\}. \quad (4)$$

Binary Term E^b. Following [17, 18], our model captures the interaction between neighboring nodes by defining the binary term E^b in Eq. 2 as:

$$E^b(Y(pq)|\{X_i\}_{i=1:n}; W_b) \stackrel{def}{=} (Y(pq) - \theta_{pq}^b f(pq))^T \Theta_{pq}^b (Y(pq) - \theta_{pq}^b f(pq)), \quad (5)$$

where $Y(pq) = [Y(p), Y(a)]$ refers to the combined pixel value for nodes p and q. $f(pq)$ is the concatenation of the integration-feature vectors $f(p)$ and $f(q)$. Similarly, parameters $\theta_{pq}^b f(pq)$ and Θ_{pq}^b represent the mean and inverse of variance of this local Gaussian model, which estimates the pixel value for a pair of neighboring nodes. Minimizing Eq. 5 will thus lead to a maximum probability of the value for two neighboring nodes as:

$$P(Y(pq)|\{X_i\}_{i=1:n}; W_b) = \exp\{-E^b(Y(pq)|\{X_i\}_{i=1:n}; W_b)\}. \quad (6)$$

Through experiments, we find the 8-neighboring connection is enough for taking into account the neighboring relations inherently held in the images, as the 24-neighboring connection slightly improves the performance but at more computational cost. Besides, this pixel-to-pixel integration mapping can be flexible to accommodate different upscaling factors.

The total energy in Eq. 2 can now be computed by summing up all local energy terms in $Y(p)$. The final HR image Y can be obtained by minimizing Eq. 2 via the L-BFGS algorithm [**?**] as:

$$\arg\min_Y E(Y|\{X_i\}_{i=1:n}, W), \quad (7)$$

In our implementation, we use the RTF [18] package[†], which provides a powerful Gaussian Conditional Random Field (GCRF) [35] model instantiation. They are widely used for image-labeling tasks, e.g., face colorization [18], image restoration [17], and non-blind deblurring [30]. Unlike existing CRF-based works, which allow

[†] http://www.gris.tu-darmstadt.de/research/visinf/software/index.en.htm

Table 1: Performance comparison of the proposed method with ten individual SR algorithms on 4 benchmarks, as well as the averaged test time (seconds per image) computed on a single core i7 4GHz CPU. Ours (Base) is trained using the 91-image training set. This table is sorted w.r.t. the computational time. The top-3 performance methods are marked in red, magenta and blue, respectively.

Upscaling Factor 3×	Set14			BD100			Urban100			ImageNet400		
	PSNR	SSIM	time(s)	PSNR	SSIM	time(s)	PSNR	SSIM	time(s)	PSNR	SSIM	time(s)
DRRN [34]	30.13	0.8349	71.19	31.24	0.8705	43.15	26.67	0.8197	60.38	30.55	0.8511	76.87
DRCN [21]	29.77	0.8317	56.31	31.04	0.8675	36.20	26.49	0.8159	43.71	30.35	0.8499	63.71
SelfEx [16]	29.35	0.8230	25.06	30.15	0.8523	20.46	25.97	0.7998	23.70	29.52	0.8366	30.63
JOR [3]	29.21	0.8205	11.08	30.05	0.8508	7.87	25.43	0.7834	6.68	29.39	0.8347	12.83
SRCNN [6]	29.14	0.8170	4.91	29.89	0.8460	4.55	25.35	0.7757	4.50	29.31	0.8298	5.71
Ours (Base)	29.55	0.8278	1.68	30.63	0.8590	1.25	25.97	0.7998	1.59	29.98	0.8423	1.85
RFL [31]	29.09	0.8197	1.34	29.95	0.8470	0.76	25.33	0.7771	1.02	29.30	0.8313	0.12
FSRCNN [7]	29.43	0.8269	0.40	30.47	0.8564	0.35	25.87	0.7952	0.37	29.77	0.8395	0.43
A+ [38]	29.27	0.8212	0.23	30.11	0.8515	0.45	25.48	0.7846	0.31	29.47	0.8354	0.80
NBSRF [29]	29.25	0.8217	0.12	30.18	0.8523	0.08	25.56	0.7870	0.06	29.51	0.8360	0.12
Bicubic	27.67	0.7769	0.0	28.52	0.8114	0.0	24.01	0.7144	0.0	27.98	0.8006	0.0

arbitrary forms of the constructed graphs, RTFs focus on image related problems and regard the input image as a probability graph.

Theoretically, we need to train our model by minimizing the differences between the combined results and the ground-truth images during the reconstruction stage. However, to reduce the training costs and to keep the size of the trained model manageable, we specify the maximum depth for the unary and binary trees to be 7 in our implementation. From our experience, a higher tree depth would not improve the performance significantly.

4 Results

4.1 Experimental Settings

Datasets. For a fair comparison with existing methods, we follow the experimental settings in SRCNN [5] [6], JOR [3]. We train our model using a 91-image training set as in [5].

For testing, we report our SR performance on three popular benchmarks, Set14 [44], BD100 (which is formed by extracting 100 images from the Berkeley Segmentation Dataest (BSD500)) [25], and Urban100 (which contains 100 challenging, highly textured urban scenes) [16]. They contain 14, 100 and 100 images, respectively. We have also tested our model using our own dataset called ImageNet400 (which is formed by randomly selected 400 images from ImageNet).

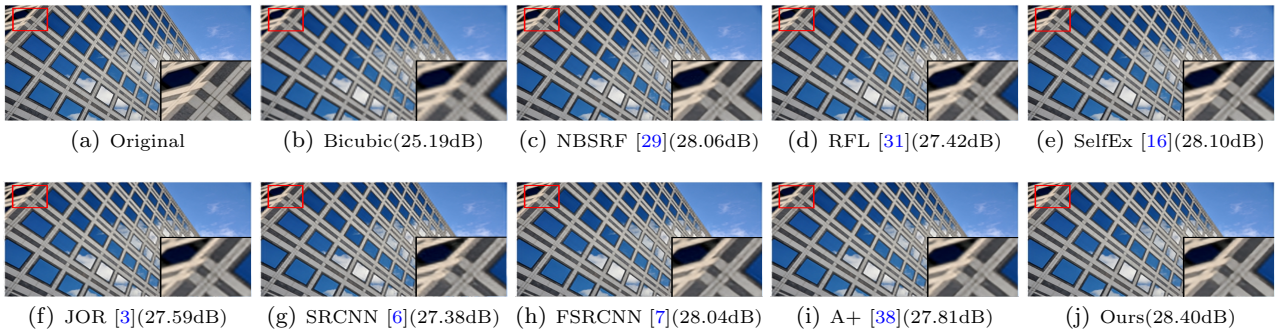
Methods for Evaluation. For a full evaluation, we choose nine state-of-the-art learning-based SR methods (i.e., A+ [38], JOR [3], SelfEx [16], RFL [31], FSRCNN [7], DRCN [21], SRCNN [6], DRRN [34] and the NBSRF [29]), as well as the Bicubic method as the candidates for comparison. The source code for these methods are publicly available from the authors, and their original parameters are set.

Table 2: Summary of the three convolutional neural network structures used. For example, Conv(3,2,64) represents that this convolutional layer has 2 input channels and 64 output channels, with a kernel size of 3x3. Each Conv layer is followed by an ReLU activation function, except the last ones. These three CNN models are trained from scratch on the 91-image training set.

	CNN-v1	CNN-v2	CNN-v3
Extraction	Conv(3,2,64)	Conv(3,2,64)	Conv(3,2,64)
Mapping	8 Conv(3,64,64)	13 Conv(3,64,64)	18 Conv(3,64,64)
Reconstruction	Conv(3,64,1)	Conv(3,64,1)	Conv(3,64,1)

4.2 Comparing to State-of-the-Art Methods

Quantitative evaluation. Table 1 compares the performance of our integration model, which is referred to as *Ours (Base)*, on the PSNR and SSIM metrics. The upscaling factors of all methods are set to 3× in this experiment. The proposed method performs consistently better than eight individual methods on all four datasets and on both evaluation metrics, while approaching the two very deep networks (i.e., DRCN [21] and DRRN [34]) with a reasonable time cost. This shows that our integration model reasonably exploits both the advantages from FSRCNN and A+ methods. It is worth noting that as both DRCN and DRRN require tens of seconds of CPU time to predict one single image, it is difficult to apply them in real-time



(a) Original (b) Bicubic(25.19dB) (c) NBSRF [29](28.06dB) (d) RFL [31](27.42dB) (e) SelfEx [16](28.10dB)

(f) JOR [3](27.59dB) (g) SRCNN [6](27.38dB) (h) FSRCNN [7](28.04dB) (i) A+ [38](27.81dB) (j) Ours(28.40dB)

Fig. 3: Challenging scene from the Urban100 dataset. The upscaling factor is set to 3×.

applications. In addition, training the DRCN [21], for example, requires days of time on a Titan X GPU, while our integration model requires only a few hours to train on the Intel i7 CPU.

Qualitative evaluation. In addition to the evaluations on accuracy and speed, visual quality from human perception on the super-resolution images are also important.

Figure 3 shows the results of one challenging urban scene with highly repetitive textures from Urban100, produced by different methods. Figure 4 shows results of some natural images from BD100 and ImageNet400. These figures illustrate that our framework generally produces higher PSNR and SSIM values as well as higher visual quality outputs among the fast SR methods, even on the challenging benchmark.

4.3 Comparing to Baseline Methods

Baselines. To further demonstrate the effectiveness of our integration model, we compare performances with one classical and widely applied strategy: estimation averaging (*AVG*). In fact, the *AVG* strategy takes the average of all inputs to compute the final result, which has the advantage of reducing the noise. However, since it deals with each input equally, some bad estimation results may compromise the good estimations.

We then perform the experiments on the integration models by using three different CNNs. Specifically, we follow the VGG-net style, which is originally designed for ImageNet classification [33], by adopting a small convolutional kernel size in order to save training parameters. We gradually increase the number of convolutional layers of the mapping part for wider receptive fields and higher level features. The network structures are summarized in Table 2 and all three models are trained from scratch on the 91-image training set.

Quantitative evaluation. Table 3 compares the performance of our integration model with the above four integration strategies. We can see that our integration model performs better on the reconstruction quality than the *AVG* baseline. Besides, while achieving similar SR performance compared with the CNN-based models, our proposed model is much faster than them (in CPU time) on the test stage (around 1 second vs. several seconds).

4.4 Delving into the Integration Strategies

To verify if our model may benefit from combining more SR methods with different algorithmic assumptions, we have trained different models for comparison, all using the 91-image training set, as shown in Table 4.

Importance of A+. We compare the integration of FSRCNN and A+ *(Ours (Base))* with other integration strategies to demonstrate the advantages of A+ as a component of our model. We regard JOR [3] and RFL [31] as exemplars of external-database-based SR methods, SelfEx [16] as an exemplar of internal-database-based method. Specifically, we replace A+ with RFL and train another combination, referred to as *FSRCNN + RFL*, which helps us verify that A+ does provide the well-estimated local priors for SR reconstruction. In contrast, without the manifold assumption, *FSRCNN + RFL* produces less satisfactory reconstruction results. Comparing the first two rows of Table 4, we can see that our model does incorporate the advantages of both FSRCNN and A+.

Are More Conventional SR Methods Helpful? We continue to explore whether our integration model *(Ours (Base))* could be benefited from SR methods of other algorithmic assumptions. We consider SelfEx [16] *(Ours (Base) + SelfEx)* as one of state-of-the-art internal-database-based SR method, and to see if

Table 3: Baseline comparison. We compare the performance of our integration model with the proposed baseline method, estimation averaging (AVG), and the three CNN-based methods, on 4 benchmarks. Bicubic is included for reference.

Upscaling Factor 3×	Set14			BD100			Urban100			ImageNet400		
	PSNR	SSIM	time(s)	PSNR	SSIM	time(s)	PSNR	SSIM	time(s)	PSNR	SSIM	time(s)
Bicubic	27.67	0.7769	0.0	28.52	0.8114	0.0	24.01	0.7144	0.0	27.98	0.8006	0.0
AVG	28.99	0.8125	0.0	29.79	0.8420	0.0	25.00	0.7607	0.0	29.15	0.8268	0.0
CNN-v1 (10 layers)	29.53	0.8276	2.55	30.53	0.8578	1.94	25.90	0.7972	2.04	29.83	0.8407	2.72
CNN-v2 (15 layers)	29.56	0.8282	3.61	30.56	0.8582	2.38	25.92	0.7977	2.96	29.87	0.8412	3.97
CNN-v3 (20 layers)	29.58	0.8278	5.12	30.55	0.8581	4.45	25.91	0.7975	4.06	29.86	0.8410	5.38
Ours (Base)	**29.55**	**0.8278**	**1.68**	**30.63**	**0.8590**	**1.25**	**25.97**	**0.7990**	**1.59**	**29.98**	**0.8423**	**1.85**

introducing internal priors could lead to a further SR performance boost. We also consider other combinations: *Ours (Base) + JOR*, *Ours (Base) + RFL*, and *Ours (Base) + All*, where *All* refers to *SelfEx + JOR + RFL*. Rows 3-6 of Table 4 show that replacing A+ with other methods or adding more methods will not significantly improve the performance, but will lead to a higher time cost. This means that A+ is sufficient to provide complementary information for FSRCNN on the SR task for variety of natural images.

Will Other Deep Learning based SR Methods be Benefited? We investigate whether this integration model can be applied to other deep learning based SR methods. To this end, we choose another state-of-the-art network (DRCN) [21] and integrate it with A+. However, the performance is generally worse than DRCN itself. This may indicate that a very deep network is not suitable for integration. Note that DRCN [21] achieves a better SR performance than SRCNN [5,6] by using a very deep network structure (from 3 to 20 layers) and by recurrently sharing filter parameters in its convolutional layers. It has more reconstruction ability and simply exceeds the upper bound of the integration model, by predicting almost all the pixels better than A+. Our integraion model fails when the performance of integrated methods vary too much. However, we actually do not consider DRCN as an integration method as it is very time-costing for online applications.

Benefits of Learning a Pixel-wise Mapping. Table 5 shows that our trained model with upscaling factor 3× can be directly applied to another upscaling factors (i.e., 4× in this example). In addition, we have also trained a model with upscaling factor 4× for comparison. Comparing the last two rows of Table 5, our model (Base-3x) achieves a competing performance to our model (Base-4x) without re-training. This demonstrates that the learned pixel-wise mapping is general for different upscaling factors, and our integration model can be efficient as we can train our model and directly apply it to different upscaling factors. Note that as learning based SR methods are typically patch-based, they can only support a specific upscaling factor, which is the factor that they are trained for. Although FSRCNN [7] provides an efficient way to handle different upscaling factors by leaving its complex mapping layers unchanged, it still needs to fine-tune its last deconvolution layer for a different upscaling factor. Due to the inherent limitation of the inputs, the proposed integration framework does not address the arbitrary upscaling factors problem. Nonetheless, we believe this pixel-wise mapping indicates a possible solution towards it.

5 Conclusion and Future Work

In this paper, we have introduced a novel integration framework to address the single image super-resolution

Table 4: Performance comparison of the proposed model, i.e., **Ours (Base)**, versus different combinations of SR algorithms from various categories on 4 benchmarks.

Upscaling Factor 3×	Set14		BD100		Urban100		ImageNet400	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ours (Base)	**29.55**	**0.8278**	**30.63**	**0.8590**	**25.97**	**0.7990**	**29.98**	**0.8423**
FSRCNN + RFL	29.51	0.8272	30.49	0.8573	25.89	0.7966	29.83	0.8405
Ours (Base) + SelfEx	29.47	0.8241	30.51	0.8546	25.89	0.7936	29.85	0.8381
Ours (Base) + RFL	29.53	0.8274	30.60	0.8585	25.94	0.7983	29.95	0.8420
Ours (Base) + JOR	29.53	0.8273	30.61	0.8583	25.93	0.7981	29.94	0.8416
Ours (Base) + All	29.57	0.8280	30.65	0.8595	25.97	0.7993	29.98	0.8427

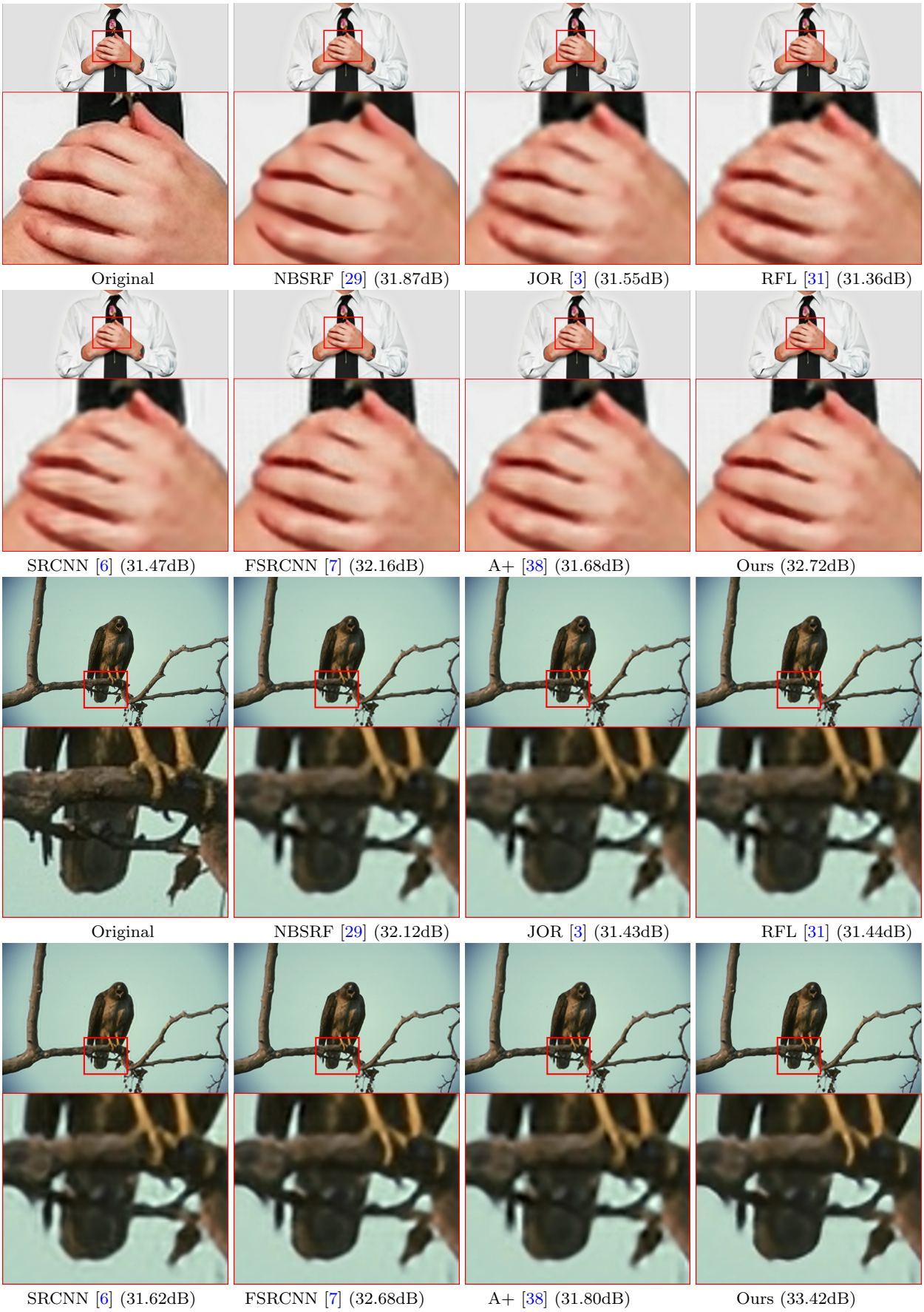


Fig. 4: The"Businessman" image (top) with upscaling factor 4× and the "bird" image (bottom) with upscaling factor 3×.

Table 5: Performance of the proposed model trained on upscaling factor 3×, applied on upscaling factor 4×, on 4 benchmarks. We refer to this model as **Ours (Base-3x)**. We have also trained the proposed model on upscaling factor 4×, referred to as **Ours (Base-4x)**, for comparison. We include the performances of other existing models trained on 4× for reference.

Upscaling Factor 4×	Set14		BD100		Urban100		ImageNet400	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
NBSRF [29]	27.42	0.7518	28.46	0.7967	23.76	0.6916	28.04	0.7889
FSRCNN [7]	27.66	0.7576	28.65	0.7993	23.94	0.6990	28.17	0.7904
RFL [31]	27.29	0.7493	28.26	0.7903	23.60	0.6808	27.80	0.7836
SRCNN [6]	27.33	0.7446	28.25	0.7882	23.57	0.6771	27.82	0.7804
JOR [3]	27.40	0.7510	28.35	0.7948	23.67	0.6872	27.88	0.7874
A+ [38]	27.46	0.7517	28.39	0.7959	23.71	0.6894	27.92	0.7886
Ours (Base-3x)	**27.80**	**0.7617**	**28.79**	**0.8042**	**24.03**	**0.7047**	**28.39**	**0.7955**
Ours (Base-4x)	**27.81**	**0.7621**	**28.84**	**0.8054**	**24.06**	**0.7059**	**28.45**	**0.7967**

problem. Our integration framework combines the advantages of a deep learning method (i.e., FSRCNN) and a classical patch-based method (i.e., A+), such that it would automatically select suitable probabilistic distributions established by these algorithms to construct the high resolution image.

We have conducted extensive evaluations on the proposed framework, both quantitatively as well as qualitatively. Our results show that the proposed framework produces optimized performances on all four popular datasets, in terms of the PSNR/SSIM metrics versus the time cost. We have also shown that our results in general have less artifacts.

As a future work, we are currently investigating two approaches to improve the performance of the proposed framework. First, we are considering the possibility of extending the integration model to multi-level, to further exploit the advantages of other SR methods. Second, we would like to design a more sophisticated memory mechanism to "remember" the characteristics/mapping of the integrated SR methods during training such that we do not need to run these SR methods during testing. It would be interesting to study if these two approaches would further improve the performance.

Acknowledgements We thank the anonymous reviewers for the insightful and constructive comments, and NVIDIA for generous donation of GPU cards for our experiments. This work is in part supported by an SRG grant from City University of Hong Kong (Ref. 7004889), and by NSFC grant from National Natural Science Foundation of China (Ref. 91748104, 61632006, 61425002, 61702194).

References

1. Chang, H., Yeung, D., Xiong, Y.: Super-resolution through neighbor embedding. In: CVPR (2004)
2. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: NIPS (2015)
3. Dai, D., Timofte, R., Van Gool, L.: Jointly optimized regressors for image super-resolution. In: EG (2015)
4. Dai, S., Han, M., Xu, W., Wu, Y., Gong, Y.: Soft edge smoothness prior for alpha channel super resolution. In: CVPR (2007)
5. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV (2014)
6. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE T-PAMI (2016)
7. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV (2016)
8. Duchon, C.: Lanczos filtering in one and two dimensions. Journal of Applied Meteorology (1979)
9. Efrat, N., Glasner, D., Apartsin, A., Nadler, B., Levin, A.: Accurate blur models vs. image priors in single image super-resolution. In: ICCV (2013)
10. Farsiu, S., Robinson, D., Elad, M., Milanfar, P.: Advances and challenges in super-resolution. International Journal of Imaging Systems and Technology (2004)
11. Fattal, R.: Image upsampling via imposed edge statistics. In: ACM TOG (2007)
12. Freedman, G., Fattal, R.: Image and video upscaling from local self-examples. ACM TOG (2011)
13. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Computer Graphics & Applications (2002)
14. Freeman, W., Liu, C.: Markov random fields for super-resolution and texture synthesis. Advances in Markov Random Fields for Vision and Image Processing (2011)
15. He, X., Zemel, R.S., Carreira-Perpiñán, M.Á.: Multiscale conditional random fields for image labeling. In: CVPR (2004)
16. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR (2015)
17. Jancsary, J., Nowozin, S., Rother, C.: Loss-specific training of non-parametric image restoration models: A new state of the art. In: ECCV (2012)
18. Jancsary, J., Nowozin, S., Sharp, T., Rother, C.: Regression tree fieldsan efficient, non-parametric approach to image labeling problems. In: CVPR (2012)

19. Kappeler, A., Yoo, S., Dai, Q., Katsaggelos, A.K.: Video super-resolution with convolutional neural networks. IEEE Transactions on Computational Imaging (2016)
20. Khler, T., Huang, X., Schebesch, F., Aichert, A., Maier, A., Hornegger, J.: Robust multiframe super-resolution employing iteratively re-weighted minimization. IEEE Transactions on Computational Imaging (2016)
21. Kim, J., Kwon Lee, J., Mu Lee, K.: Deeply-recursive convolutional network for image super-resolution. In: CVPR (2016)
22. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
23. Kim, K., Franz, M., Schölkopf, B.: Kernel hebbian algorithm for single-frame super-resolution. In: ECCV Workshop on Statistical Learning in Computer Vision (2004)
24. Kim, K.I., Kwon, Y.: Single-image super-resolution using sparse regression and natural image prior. IEEE TPAMI (2010)
25. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
26. Michaeli, T., Irani, M.: Nonparametric blind super-resolution. In: ICCV (2013)
27. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. IEEE Signal Processing Magazine (2003)
28. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science (2000)
29. Salvador, J., Perez-Pellitero, E.: Naive bayes super-resolution forest. In: ICCV (2015)
30. Schmidt, U., Rother, C., Nowozin, S., Jancsary, J., Roth, S.: Discriminative non-blind deblurring. In: CVPR (2013)
31. Schulter, S., Leistner, C., Bischof, H.: Fast and accurate image upscaling with super-resolution forests. In: CVPR (2015)
32. Shen, X., Tao, X., Gao, H., Zhou, C., Jia, J.: Deep automatic portrait matting. In: ECCV (2016)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
34. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR (2017)
35. Tappen, M., Liu, C., Adelson, E., Freeman, W.: Learning gaussian conditional random fields for low-level vision. In: CVPR (2007)
36. Thévenaz, P., Blu, T., Unser, M.: Image interpolation and resampling. Handbook of medical imaging, processing and analysis (2000)
37. Timofte, R., De, V., Van Gool, L.: Anchored neighborhood regression for fast example-based super-resolution. In: ICCV (2013)
38. Timofte, R., De Smet, V., Van Gool, L.: A+: Adjusted anchored neighborhood regression for fast super-resolution. In: ACCV (2014)
39. Wang, Q., Tang, X., Shum, H.: Patch based blind image super resolution. In: ICCV (2005)
40. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: ICCV (2015)
41. Xie, Z., Xu, K., Liu, L., Xiong, Y.: 3d shape segmentation and labeling via extreme learning machine. In: Computer Graphics Forum (2014)
42. Xu, K., Shi, Y., Zheng, L., Zhang, J., Liu, M., Huang, H., Su, H., Cohen-Or, D., Chen, B.: 3d attention-driven depth acquisition for object identification. ACM TOG (2016)
43. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. IEEE TIP (2010)
44. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces (2010)