Infrared Physics and Technology 105 (2020) 103193

Contents lists available at ScienceDirect

Infrared Physics & Technology

journal homepage: www.elsevier.com/locate/infrared

Image super-resolution based on half quadratic splitting

Kuanhong Cheng, Juan Du, Huixin Zhou*, Dong Zhao, Hanlin Qin

School of Physics & Optoelectronic Engineering, Xidian University, Xi'an 710071, China

ARTICLE INFO

Keywords:
Image restoration
Single image super-resolution
Half quadratic splitting
Guided image filter

ABSTRACT

Model-based optimization methods have been widely used in varies image restoration solutions and achieved some remarkable results. However, finding out a closed mathematical solution for certain priors remains a great challenge. To resolve this problem, this paper presents an improved model-based algorithm for single image super-resolution. Instead of focusing on specific prior knowledge, we exploit the optimization scheme of general image restoration formula. In our approach, the general format of model-optimization problem is transformed into an alternant renewal process through half quadratic splitting. This transform can also separate the optimization into a modular structure and allows us to optimize the fidelity term and regularization term separately. Of which the regular optimization process can be considered as a denoising process. Then the guided filter is taken as a denoiser to realize this optimization, which uses local linear transform to keep the detail and L2 norm constraint to smooth the noise of input image. Experiments with benchmark datasets and our own infrared images show that our method can surpass several famous model-based and data-based methods in PSNR and SSIM.

1. Introduction

Image super-resolution (SR) intends to recover high-resolution (HR) clear images from low-resolution (LR) or blurred input images through software algorithms, which has become one of the most popular research topics of low-level image processing problem because of its huge potential to improve the quality of images at a very low cost.

Current single image SR methods can be divided into three categories: interpolation-based methods, model-based methods, and data-based methods.

Interpolation methods generate new pixels from neighboured pixels through interpolation (such as bicubic interpolation, cubic spline interpolation [1], and Edge-directed interpolation [2]) or local image filtering, which are simple in theory and easy to be conducted but can not get satisfied performance due to the shortage of useful information.

Data-based methods such as CNNs have recently become a hot point for image processing and achieved excellent results in many high-level vision problems such as object detection [3] and image inpainting.[4] But for image restoration such as SR, the model is always trained by a dataset with several certain degradation matrices, which makes it not suitable to tackle different down-sample or blur kernels. (for example, the model trained in [5] has an excellent performance on the image with the same scale factor as the training set, but degrades rapidly on the dataset with different down-scale images).

Model-based methods try to design an objective function based on reasonable assumptions or prior knowledge about the observation model that maps the HR image to the LR ones, then minimize this function to get results. This kind of scheme can be applied to images with any degradation models and achieve different purposes include denoising, deblurring, and SR. Thus has become one of the most studied approaches for decades. Various priors have been proposed to regularize the SR restoration, most of the recent researches focus on two kinds of models: sparse representation [6] and low-rank [7] model.

Sparse representation comes from the well-known compressed sensing (CS) [8], which claims that any natural image can be considered as a linear representation with respect to a set of sparse and over complete atoms. Jianchao Yang et al. [9] introduced a patch-based sparse representation algorithm for image SR, where they first construct two dictionaries D_h and D_l for HR and LR images. Then for each input LR patch y, find a sparse representation with respect to D_l. The corresponding HR patches D_h will be combined according to these coefficients to generate the output high-resolution patch x. After that, take the reconstructed HR into the degraded model to minimize the error between the degraded image and input LR image to get the final result. Based on this method, Roman Zeyde et al. [10] presented an improved sparse-land SR algorithm. Instead of training two dictionaries, their LR dictionary is generated through a scale-down operator, a pre-processing is also applied to each patch pairs to remove the low-frequencies of HR

* Corresponding author.
E-mail address: hxzhou@mail.xidian.edu.cn (H. Zhou).

https://doi.org/10.1016/j.infrared.2020.103193
Received 20 August 2019; Received in revised form 31 December 2019; Accepted 10 January 2020
Available online 13 January 2020
1350-4495/

patches. Dimensionality reduction technology is also to accelerate the dictionary training. These modifications help to reduce the compute complexity as well as gain performance.

Sparse representation has achieved some promising results for SR problem [11,12,13]. But this patch-based prior can not fully utilize the similarity among different patches, and also need to train dictionary from a huge number of image pairs, which can not be obtained from single input LR image. Besides, it may even introduce noise or irrelevant data that can degrade the subsequent reconstruction. To resolve this problem, Chih-Yuan Yang et al. [14] exploited the self-similarity of different patches within the input LR image. Where the training patches are generated directly from the input image itself. Thus no external information is required for dictionary training, which also makes it more practical for real applications. The performance of this method is also better than the original sparse representation method in [9]. Jiaying Liu et al [15] then combine the above internal and external information together through a feature generation strategy. In their algorithm, the atoms of sparse representation come from two resources. First, a content-based retrieval technique is applied to the external dataset to select correlated images that have similar features as the input image. Then external patches are extracted and refined due to the content of input LR patches. After that, the internal patches are obtained in the same way as the self-similarity method. A dictionary is constructed as a combination of these internal and external patches. The SR results are finally obtained as a representation of this dictionary. Because of the fully utilize of self-similarity and external information, this approach has a better SR performance with the criterion function peak signal to noise ratio (PSNR).

Both sparse representation and self-similarity can be considered as dictionary-based methods, they try to learn a group of atoms from an external dataset or several internal patches. This kind of method uses linear representation to reconstruct HR images that utilize the sparse attribute of natural signals. Recently, some researchers try to use the intrinsic low-rank attribute among correlated patches within the input image [16,17,18]. The idea also comes from self-similarity, but the optimization is totally different. Instead of finding atoms and representation coefficients, the low-rank model format those similar patches as a matrix with each column being a stretched patch vector. Then realize SR by minimizing the rank of this matrix.

In this paper, we present a novel scheme for image SR which uses guided image filter (GIF) [20] and half quadratic splitting (HQS) [19]. The highlights of our scheme are listed as follows.

(1) HQS is used to split the optimize of the data-fitting term and regularization term. This is inspired by the fact that in many cases, it is very hard to find a closed mathematical solution for specific prior. So we introduce an auxiliary variable to transform the optimization into a two-phase iteration process.
(2) GIF is taken as a Gaussian denoiser to update the SR result from the auxiliary variable.
(3) Experimental results with both benchmark datasets and infrared images show that our algorithm outperforms several famous model-based and data-based methods.
(4) The rest of this paper is organized as follows. We first discuss the HQS and GIF algorithm in Section 2, The proposed method is described in Section 3. Experimental results and discussion are presented in Section 4. Followed by the conclusion in Section 5.

2. Half quadratic splitting and guided image filter

2.1. Half quadratic splitting

Model-optimization provides a new solution for many ill-posed or NP-hard problems that can not be solved by strict mathematical methods. The general format of model-optimization can be written as

$$E = f(\theta, x, y) + g(x) \tag{1}$$

where E means the objective function, x is the signal we want to reconstruct, y represents the observation. Data-fitting term f describes the relationship between observation y and the ideal signal $\times$, with parameters θ. regularization term g measures prior knowledge.

The solution of Eq. (1) sometimes is hard to find, thus several variable separating methods such as HQS, ADMM have been proposed to resolve this matter. With the help of HQS, Eq. (1) can be rewritten as follows

$$E = f(\theta, x, y) + g(z),\ s.\ t.\ z = x \tag{2}$$

As can be seen in Eq. (2), with a new auxiliary z, the data-fitting and regularization terms have been successfully separated since there is no common variable in these two terms. To solve Eq. (2), one most used scheme is to add another term to measure and minimize the error between z and $\times$ as Eq. (3).

$$L = f(\theta, x, y) + g(z) + d(z, x) \tag{3}$$

where μ is the weighted value, $d(z, x)$ is the distance between z and x. Eq. (3) can be solved by an alternant renewal process for $\times$ and z respectively as

$$x_{n+1} = \mathrm{argmin}(f(\theta, x, y) + d(z_n, x)) \tag{4}$$

$$z_{n+1} = \mathrm{argmin}(g(z) + d(z, x_{n+1})) \tag{5}$$

As can be seen in Eq. (4) and Eq. (5), x and z can be updated through minimizing the terms which contain corresponding variables. The solution of these formulas can be obtained by partial differential method and gradient descent scheme.

2.2. Guided image filter

The GIF requires two parameters: input image and guidance image, and can adjust filter coefficients due to the details of guidance image. Thus the result can be considered as a fusion of the details of guidance image and contours of the input image.

There are two key points of this algorithm: the assumption and constraint criterion.

First, GIF assumes a piecewise linear transform between guidance and output images, which is used to ensure the output image would have the same details as guidance image. This relationship can be expressed as

$$q_i = a_k I_i + b_k,\ \forall\ i \in w_k \tag{6}$$

where q is output image, I means guidance image, w_k is the k_{th} local window of these two images, i represents the index of pixels within this local window. a_k and b_k are transform coefficients which are assumed to be constant within w_k.

The second point is the constraint. The GIF want to construct an image with the same contours as input image, therefore the objective function is set as the $L2$ norm between input and output images as

$$a_k, b_k = \mathrm{argmin}\left(\sum_{i \in w_k} ((a_k I_i + b_k - p_i)^2 + \varepsilon a_k^2)\right) \tag{7}$$

Here εa_k^2 is a regularization term. Eq. (7) can be solved through linear regression as

$$a_k = \frac{\frac{1}{|w|}\sum_{i \in w_k} I_i p_i - \mu_k \overline{p}_k}{\sigma_k^2 + \varepsilon} \tag{8}$$

$$b_k = \overline{p}_k - a_k \mu_k \tag{9}$$

where μ_k and σ_k^2 are the mean value and variance of local window w_k in I, $|w|$is the number of pixels within w_k, $\overline{p}_k$ is the mean value of p in w_k.

Because of the overlapping of local windows, each pixel can be involved in several different local patches with individual coefficients. Thus the ultimate transform coefficients of i are finally determined by

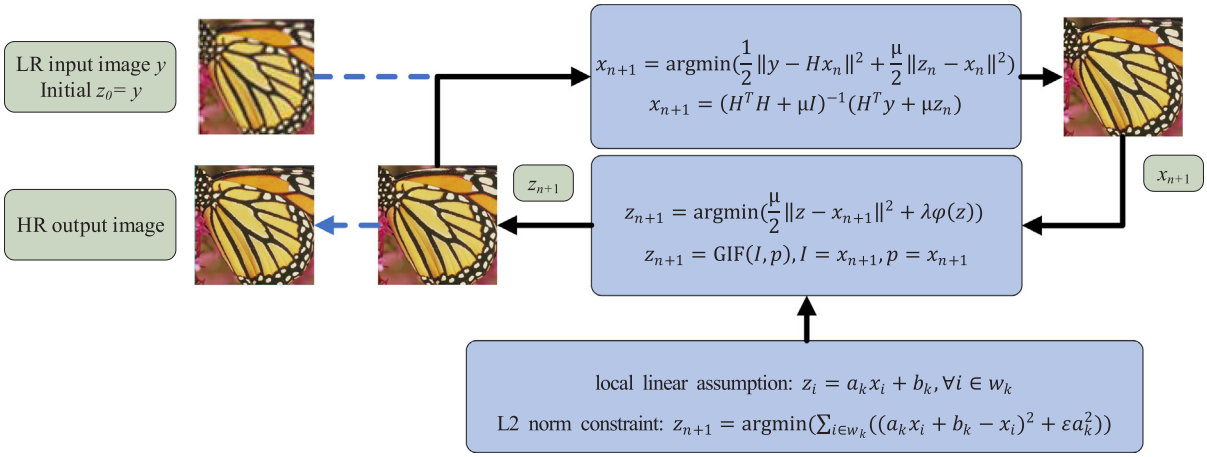


Fig. 1. Flowchart of HQS and GIF based image super-resolution.

an average filter with the same window size in Eq. (6).

$$\bar{a}_i = \frac{1}{|w|}\sum_{k \in w_i} a_k \tag{10}$$

$$\bar{b}_i = \frac{1}{|w|}\sum_{k \in w_i} b_k \tag{11}$$

The output image is then obtained by the linear transform of guide image as in Eq. (6).

$$q_i = \bar{a}_i I_i + \bar{b}_i \tag{12}$$

GIF has been widely used in various applications, mostly it is taken as an edge-preserve image smoothing filter [21,22,23]. It is used in our method to conduct the estimation of z in Eq. (5), details will be discussed in sector 3.

3. Image super-resolution based on hqs and guided image filter

The general format of model-based optimization method and HQS scheme have been analyzed in chapter 2.1. For image restoration, some of the parameters in Eqs. (1)–(5) shall be specified.

Image restoration tries to reconstruct a high-quality image from a degraded observation. As a well-known ill-posed problem, one most used method is to estimate the output through the maximum posterior probability (MAP) approach. Which assumes that for an observation y, we can get an estimation of ideal image $\hat{x}$ through Bayesian frame as

$$\hat{x} = \text{argmax}(p(y|x)p(x)) \tag{13}$$

where p is probability and $|$ donates conditional probability. Eq. (13) is equivalent to Eq. (1) for the solution.

However, in Eq. (13), $p(x)$ always remains unknown, and for most blind image restoration, $p(y|x)$ is also undiscovered. Thus additional prior must be imposed to get the solution. Most image restoration problems can be modeled by a degenerated matrix and addictive noise as $y = Hx + n$, where n represents noise.

Due to the above description, the parameter θ of data-fitting term $f(\theta, x, y)$ is a degenerate matrix. For image SR problem, it is a cyclic matrix of down-sample kernel H. The data-fitting term tries to measure the difference between $\times$ and degenerated observation y which can be described by $\|y - Hx\|^2$.

The regularization term $g(x)$ represents the prior of ideal SR images. But the final simplified formula shows that this term can remain unknown, we use $\varphi(x)$ to represent it.

We use $L2$ norm to measure the difference between $\times$ and auxiliary parameter z. Therefore the new penalty term in Eq. (3) is $\|z - x\|^2$.

Due to the above analysis, the corresponding format of Eq. (3) for SR can be rewritten as

$$L = \frac{1}{2}\|y - Hx\|^2 + \lambda\varphi(z) + \frac{\mu}{2}\|z - x\|^2 \tag{14}$$

where λ and μ are weighted valus for regularization term and additional penalty term. The solution of Eq. (4) and Eq. (5) are

$$x_{n+1} = \text{argmin}\left(\frac{1}{2}\|y - Hx\|^2 + \frac{\mu}{2}\|z_n - x\|^2\right) \tag{15}$$

$$z_{n+1} = \text{argmin}\left(\frac{\mu}{2}\|z - x_{n+1}\|^2 + \lambda\varphi(z)\right) \tag{16}$$

Which can be solved by gradient descent scheme.

Note that Eq. (16) can also be considered as a Gaussian noise removal process, which means z can be updated by performing denoising algorithm on x. GIF is utilized here as a denoiser to achieve the update.

$$z_{n+1} = \text{GIF}(I, p),\ I = x_{n+1},\ p = x_{n+1} \tag{17}$$

where I and p are guidance image and input image of GIF as in Eqs. (6)–(12). The assumption and constraint of GIF for the proposed SR algorithm are defined as

$$z_i = a_k x_i + b_k,\ \forall\ i \in w_k \tag{18}$$

$$z_{n+1} = \sum_{i \in w_k}((a_k x_i + b_k - x_i)^2 + \varepsilon a_k^2) \tag{19}$$

After the discussion of the model and parameters HQS and GIF, the flowchart of our proposed method is presented in Fig. 1.

Our SR algorithm consists of several main steps. First, the input image is upsampled through bicubic interpolation to generate an HR input image y and initialize the output image z. After that, x is updated by minimizing data-fitting error through PDE scheme as Eq. (15), which turns out to be a pseudo-inverse problem. Followed by the GIF performed on $\times$ to update z as in Eq. (18) and Eq. (19). The updated z is then taken as the input of the next iteration to refine the construction. The value of z in the last iteration is finally obtained as SR result.

4. Experimental results and analysis

The performance of the proposed method is first tested on three different datasets: Set5, Set14, and B100, which have been widely used as benchmarks for image restoration algorithm. Our experiments evaluate the proposed algorithm and compare it with the other 7 typical model-based SR methods from literature. And since deep learning has been widely applied to image restoration [30,31] and super resolution [29,32], we also added two deep learning methods for comparison. They are:

(1) The classic bicubic interpolation method.
(2) Zeyde [10]. Which is an improved sparse-representation based SR

algorithm, with a pre-processing scheme applied to each patch pairs to remove the low-frequencies of HR patches.

(3) Global regression (GR) [24]. This method supports the usage of sparse learned dictionaries in combination with embedding methods. The solution is obtained by global regression.

(4) Anchored neighborhood regression (ANR) [24]. ANR means to anchor the neighborhood embedding of a low-resolution patch to the nearest atom in the dictionary and to precompute the corresponding embedding matrix. In this case, the nearest neighbors are computed using the correlation with the dictionary atoms rather than the Euclidean distance.

(5) Neighbor embedding with least squares (NE + LS). This algorithm use neighbor embedding as an assumption with least square error as a constraint.

(6) Neighbor embedding with locally linear embedding (NE + NNLS) [25]. It uses a dictionary of low resolution (LR) and high resolution (HR) trained patch pairs to infer the unknown HR details. Each LR feature vector in the input image is expressed as the weighted combination of its K nearest neighbors in the dictionary; the corresponding HR feature vector is reconstructed under the assumption that the local LR embedding is preserved.

(7) Neighbor embedding with locally linear embedding (NE + LLE) [26]. Inspired by the famous manifold learning methods locally linear embedding (LLE) [27]. This algorithm assumes that small image patches in the low- and high-resolution images form manifolds with similar local geometry in two distinct feature spaces. And tries to reconstruct feature vectors corresponding to image patches by their neighbors in the feature space.

(8) SRGAN [28]. It is a deep-learning method which uses GAN and perceptual loss to achieve SR.

(9) SRCNN [32]. The classical deep-learning based single image SR method.

In the proposed algorithm, the local window size of GIF is 3×3 with $\varepsilon = 1e-3$ in Eq. (19), the total iteration number is 20. All the experiments are conducted on a Windows 10 PC with Matlab R2018b. Scale factors are set from x2 to x4. PSNR values are calculated from the Y channel of *YCrCb* space as commonly used in other papers.

We first summarize the average PSNR and SSIM of 3 datasets with 3 different scale factors (x2, x3, x4) in Table 1. From which we can see that among these 9 group datasets, our algorithm achieves the highest PSNRs in 8 groups, which surpasses all the other methods and shows a very good performance. But we also find that our method can only achieve 1 highest SSIM, SRCNN obtains 7 best SSIM scores which is the best one. We think the reason is that SSIM is a high-level perceptual evaluation, not a pixel-wise score like PSNR, and the deep learning method has a large perceptual field through the multi-layer convolutional operation. As a traditional model-based method, our algorithm involves only local operations such as the guided filter with filter size 3×3, which can not cover such field.

We also compare the PSNR and SSIM for each input image and scale. The results of each image in Set5 with scale factor x2, x3 and x4 are summarized in Table 2. As can be seen, our proposed method gets 8 highest PSNR and 3 highest SSIM out of 15 experiments, which is the first and second respectively. Followed by ANR with 7 best PSNR and 4 SSIM results. Even for those images which our algorithm can not achieve the top performance, it is very close to the best. We also found that SRCNN outperforms all the dictionary-based methods in SSIM with 6 best scores but failed in PSNR. This is mainly caused by its powerful perceptual field which can not be achieved by traditional algorithms.

In order to compare different algorithms through visual assessment, we present the SR results of the butterfly image in Set5 with scale factor x2 in Fig. 2.

As can be seen in Fig. 2, the bicubic SR result suffers heavily from blurring problem, which is caused by the excessively simple model that can not fully utilize the useful information. Zeyde's method (d) achieves great improvement compared to the classic bicubic approach, but still can not compete with our proposed algorithm in detail reconstruction. The common problem of dictionary methods and NE methods is the artifacts introduced through the construction, which can be noticed by naked eyes from the enclosed area in (e) GR, (f) ANR, (g) NE + NNLS and (i) NE + LLE as some ringing textures. Besides, the details of compared methods (c) bicubic, (e) GR and (f) ANR also can not compete with the proposed method since the edges of the proposed method's result are sharper and cleaner than that of these algorithms, which can be distinguished through the boundary of different color areas. The SGRAN and SRCNN results (j, k) have better visual performance than those dictionary methods in structure and details but still have very slight artifacts. Therefore, the visual assessment shows that our method has the best performance in both detail recovery and artifacts suppression.

The PSNR of each image in Set14 with scale factor x3 are shown in Table 3, with SR results of the 10 algorithms of the pepper image presented in Fig. 3.

Table 3 shows the proposed method gets 8/14 of the top PSNRs but only 1/14 of SSIMs, which is similar to the above results in Tables 1 and 2.

Fig. 3(c) indicates a very bad performance of the classic bicubic algorithm. (d) Zeyde, (f) ANR, (g) NE + LS, (h) NE + NNLS and (i) NE + LLE have more details, but still less than the proposed results shown in (l). The artifacts in (e) GR are very serious around the border of different color peppers. Some ringing can also be found in (f) ANR, (i) NE + LLE, SRGAN (j), and SRCNN (k) by visual identification. Whereas the result of the proposed method has no ringing artifacts, and the detail of edges is the best among all the methods.

We also test our proposed method on infrared images. All the images in this experiment are taken by an uncooled long-wave infrared (LWIR) camera with resolution 320×240. The experiment is conducted on 5 images with scale factors x2, x3, and x4. Input images are shown in Fig. 4. Table 4 presents the PSNR and SSIM, with SR reconstructed images presented in Fig. 5.

Table 4 indicates our method achieved almost all the best results in

Table 1
Average PSNR/SSIM of 3 benchmark datasets.

Dataset	Scale	Bicubic	Zeyde	GR	ANR	NE + LS	NE + NNLS	NE + LLE	SRGAN	SRCNN	Proposed
Set5	x2	33.7/0.914	**35.8**/0.936	35.1/0.927	**35.8**/0.936	35.7/0.935	35.4/0.933	**35.8**/0.935	34.2/0.935	35.3/**0.943**	35.7/0.927
	x3	30.4/0.855	31.9/0.880	31.4/0.866	31.9/0.879	31.8/0.878	31.6/0.874	31.8/0.878	30.1/0.868	31.4/**0.895**	**32.2**/0.891
	x4	28.4/0.805	29.7/0.829	29.3/0.815	29.7/0.828	29.6/0.826	29.5/0.823	29.6/0.827	28.8/0.834	29.2/0.843	**30.0**/**0.851**
Set14	x2	30.2/0.853	31.8/0.882	31.4/0.878	31.8/0.883	31.7/0.880	31.6/0.878	31.8/0.882	30.4/0.889	31.3/**0.894**	**32.0**/0.863
	x3	27.5/0.765	28.7/0.793	28.3/0.787	28.7/0.794	28.6/0.791	28.4/0.787	28.6/0.793	26.9/0.791	28.2/**0.812**	**28.9**/0.793
	x4	26.0/0.702	26.9/0.726	26.6/0.719	26.9/0.727	26.8/0.723	26.7/0.720	26.8/0.725	25.6/0.734	26.4/**0.740**	**27.2**/0.732
B100	x2	29.3/0.821	30.4/0.852	30.2/0.854	30.4/0.855	30.4/0.851	30.3/0.850	30.4/0.854	30.1/**0.883**	29.6/0.882	**30.8**/0.865
	x3	27.2/0.736	27.9/0.762	27.7/0.761	27.9/0.765	27.8/0.761	27.7/0.758	27.9/0.764	27.0/0.769	27.0/**0.789**	**28.1**/0.776
	x4	25.9/0.678	26.5/0.699	26.4/0.698	26.5/0.701	26.5/0.698	26.4/0.696	26.5/0.700	25.7/0.699	25.5/**0.710**	**26.7**/0.707

Table2
PSNR/SSIM of Set5 with 3 different scales.

Scale	Image	Bicubic	Zeyde	GR	ANR	NE + LS	NE + NNLS	NE + LLE	SRGAN	SRCNN	Proposed
x2	baby	37.1/0.950	38.2/0.960	38.3/0.962	**38.4**/0.962	38.1/0.959	38.0/0.957	38.3/0.960	36.4/0.952	37.2/**0.963**	37.8/0.938
	bird	36.8/0.969	39.9/0.981	39.0/0.978	**40.0**/**0.982**	39.9/0.980	39.4/0.978	40.0/0.981	37.2/0.971	39.5/0.980	39.5/0.933
	butterfly	27.4/0.863	30.6/0.915	29.1/0.877	30.5/0.912	30.4/0.915	30.0/0.909	30.4/0.911	31.5/**0.958**	31.4/0.956	**32.1**/0.916
	head	34.9/0.852	35.6/0.870	35.6/0.873	**35.7**/0.872	35.5/0.870	35.5/0.868	35.6/0.871	31.8/0.829	34.4/0.845	35.2/**0.894**
	woman	32.1//0.934	**34.5**/0.952	33.7/0.946	**34.5**/0.953	34.3/0.951	34.2/0.950	34.5/0.953	33.8/0.953	34.0/**0.969**	34.2/0.954
x3	baby	33.9/0.909	**35.1**/0.922	34.9/**0.923**	**35.1**/0.922	35.0/0.920	34.8/0.918	35.1/0.922	32.5/0.889	33.9/0.919	34.9/0.908
	bird	32.6/0.928	34.6/0.946	33.9/0.939	34.6/**0.947**	34.4/0.944	34.2/0.942	34.6/**0.947**	31.7/0.914	34.1/0.946	**34.7**/0.912
	butterfly	24.0/0.764	25.9/0.814	25.0/0.763	25.9/0.807	25.8/0.815	25.6/0.803	25.7/0.807	26.7/0.892	26.6/**0.896**	**27.5**/0.873
	head	32.9/0.795	**33.6**/0.812	33.5/0.813	**33.6**/0.816	33.5/0.812	33.5/0.810	33.6/0.814	30.0/0.745	32.4/0.782	33.4/**0.851**
	woman	28.6/0.879	**30.4**/0.903	29.7/0.890	30.3/0.906	30.2/0.901	29.9/0.896	30.2/0.902	29.6/0.908	30.0/**0.929**	**30.4**/0.910
x4	baby	31.8/0.870	**33.1**/0.887	32.8/0.887	33.0/**0.889**	32.9/0.885	32.8/0.883	33.0/0.887	31.4/0.849	31.8/0.869	**33.1**/0.874
	bird	30.2/0.883	31.7/0.903	31.3/0.895	31.8/0.901	31.6/0.900	31.5/0.898	31.7/0.903	30.3/0.878	31.2/**0.897**	**32.1**/0.884
	butterfly	22.1/0.691	23.6/0.732	23.1/0.686	23.5/0.723	23.4/0.726	23.3/0.729	23.4/0.721	**24.7**/**0.854**	24.1/0.836	**24.7**/0.823
	head	31.6/0.752	32.2/0.769	32.1/0.768	**32.3**/0.771	32.2/0.768	32.1/0.766	32.2/0.770	29.8/0.712	31.1/0.730	32.2/**0.813**
	woman	26.5/0.830	27.9/0.854	27.4/0.839	27.8/0.852	27.6/0.850	27.6/0.846	27.7/0.851	**28.0**/0.878	27.6/**0.881**	**28.0**/0.858

(a) Original image (b) Down-sample image (c) Bicubic/27.4 (d) Zeyde/30.6 (e) GR/29.1 (f) ANR/30.5

(g) NE+LS/30.4 (h) NE+NNLS/30.0 (i) NE+LLE/30.4 (j) SRGAN/31.5 (k)SRCNN/31.4 (l) Proposed/32.1

Fig. 2. SR results of different algorithms on the butterfly image in Set5 with scale factor x2.

Table3
PSNR/SSIM of Set14 with scale x3.

Image	Bicubic	Zeyde	GR	ANR	NE + LS	NE + NNLS	NE + LLE	SRGAN	SRCNN	Proposed
baboon	23.2/0.524	23.5/0.561	23.5/0.571	**23.6**/0.569	23.5/0.562	23.5/0.560	**23.6**/0.568	21.5/0.553	22.3/**0.589**	23.5/0.552
barbara	26.20.759	**26.8**/0.778	26.8/**0.780**	26.7/0.777	26.7/0.776	26.7/0.774	26.7/00778	24.6/0.743	25.2/0.769	26.6/0.762
bridge	24.4/0.634	25.0/0.673	24.9/0.678	25.0/0.676	24.9/0.669	24.9/0.666	25.0/0.674	24.8/0.694	**26.6**/**0.787**	26.4/0.728
coastguard	26.6/0.626	**27.1**/0.663	27.0/**0.667**	**27.1**/0.665	27.0/0.659	27.0/0.657	**27.1**/0.664	25.9/0.655	26.0/0.664	**27.1**/0.635
comic	23.1/0.680	24.0/0.725	23.8/0.715	24.0/0.729	23.9/0.722	23.8/0.717	24.0/0.726	23.2/0.786	23.2/**0.803**	**24.2**/0.740
face	32.8/0.794	33.5/0.812	33.5/0.813	**33.6**/0.815	33.5/0.811	33.5/0.810	**33.6**/0.813	32.5/0.836	32.4/0.782	33.4/**0.853**
flowers	27.2/0.794	28.4/0.823	28.1/0.815	28.5/0.825	28.3/0.820	28.2/0.817	28.4/0.822	27.0/0.814	27.9/**0.847**	**28.9**/0.790
foreman	31.2/0.897	33.2/0.921	32.3/0.906	33.2/0.921	33.2/0.921	32.9/0.916	33.2/0.921	29.8/0.910	32.5/**0.928**	**33.7**/0.920
lenna	31.7/0.848	33.0/0.865	32.6/0.862	**33.1**/**0.867**	33.0/0.865	32.8/0.863	33.0/0.866	30.5/0.804	32.3/0.830	33.0/0.859
man	27.0/0.741	27.9/0.769	27.6/0.763	27.9/0.771	27.9/0.767	27.7/0.764	27.9/0.770	27.0/0.775	27.1/**0.807**	**28.1**/0.804
monarch	29.4/0.906	31.1/0.923	30.4/0.912	31.1/0.921	30.9/0.921	30.8/0.918	30.9/0.920	31.5/0.930	31.5/**0.940**	**32.2**/0.924
pepper	32.4/0.857	34.1/**0.871**	33.2/0.864	33.8/0.870	33.9/0.870	33.6/0.867	33.8/0.870	29.3/0.777	33.4/0.824	**34.3**/0.837
ppt3	23.7/0.869	25.2/0.893	24.6/0.860	25.0/0.884	25.1/0.891	24.8/0.879	24.9/0.884	25.6/0.918	**25.8**/**0.928**	25.4/0.867
zebra	26.6/0.782	28.5/0.825	27.9/0.817	28.4/0.826	28.3/0.821	28.1/0.816	28.3/0.824	27.3/0.829	28.0/**0.867**	**28.7**/0.831

PSNR except Img 3 of scale x3.

Fig. 5 shows that all the dictionary-based methods and neighbor embedding methods can improve the quality of SR results compared to the basic bicubic interpolation. As they have more details can be noticed by visual assessment. But the proposed method obtains the best detail preserving performance when comes to the edges of the table and leg in the image.

In summary, these above-mentioned experimental results indicate that the proposed HQS and GIF based algorithm has a better performance of image SR for both RGB and IR images in the aspects of both PSNR evaluation and subjective visual. But we also noticed that its SSIM can not suppress some deep-learning-based methods.

We also have an evaluation of the running time of different methods on benchmark datasets with different scale factors, the results are presented in Table 5.

Since all the dictionary-based methods in our experiment share the same dictionary (with 1024 atoms), the training time of this dictionary has not been included in the running time, thus the running time of

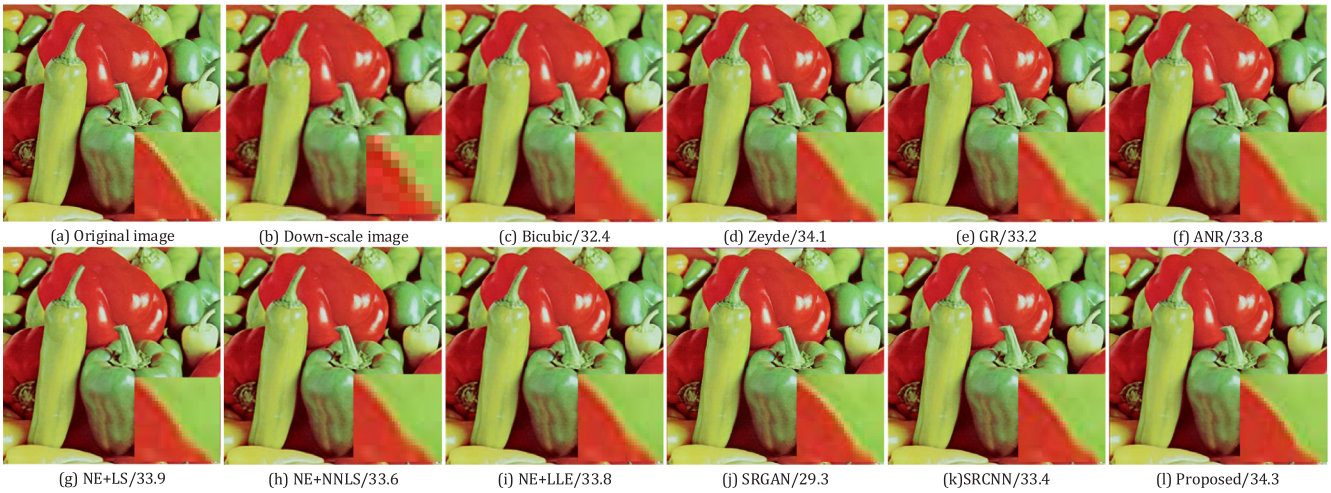


Fig. 3. SR results of different algorithms on the pepper image in Set14 with scale factor x3.

Fig. 4. LWIR images in our experiment.

Table4
PSNR/SSIM of infrared images with 3 different scales.

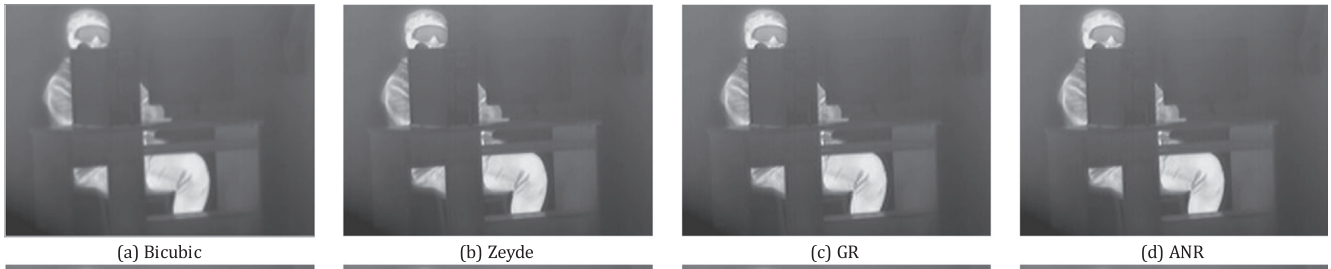
Scale	Image name	Bicubic	Zeyde	GR	ANR	NE + LS	NE + NNLS	NE + LLE	Proposed
x2	Img1	35.9/0.951	37.6/0.962	37.2/0.960	37.6/0.962	37.5/0.961	37.4/0.960	37.6/0.962	**38.0**/**0.965**
	Img2	39.6/0.979	42.1/0.984	41.2/0.984	41.9/**0.985**	41.8/0.984	41.6/0.983	41.9/0.984	**42.6**/0.983
	Img3	32.9/0.878	**33.9**/0.902	33.8/0.902	**33.9**/**0.905**	33.8/0.901	33.8/0.900	**33.9**/0.903	**33.9**/0.894
	Img4	36.4/0.961	38.6/**0.969**	37.7/0.970	38.3/**0.970**	38.3/0.968	38.0/0.968	38.3/0.969	**39.6**/0.953
	Img5	38.9/0.969	**41.2**/0.978	40.3/0.978	41.0/0.978	41.1/0.977	41.0/0.958	41.1/**0.978**	**41.2**/**0.981**
	Average	36.7/0.948	38.7/0.959	38.1/0.959	38.6/**0.960**	38.5/0.958	38.4/0.958	38.5/0.959	**39.1**/0.955
x3	Img1	32.8/0.920	34.2/0.934	33.6/0.927	34.0/0.931	34.0/0.931	33.7/0.928	34.0/0.930	**34.3**/**0.940**
	Img2	36.0/0.945	38.2/0.972	37.0/0.968	37.8/0.971	37.8/0.971	37.4/0.969	37.7/0.971	**38.9**/**0.981**
	Img3	30.7/0.820	**31.6**/0.837	31.4/**0.840**	31.5/**0.840**	31.4/0.836	31.4/0.835	31.5/0.839	31.5/0.834
	Img4	33.5/0.944	35.3/**0.951**	34.4/0.947	35.1/0.950	35.1/0.950	34.8/0.950	35.0/**0.951**	**36.1**/0.939
	Img5	35.7/0.956	37.9/0.962	36.7/0.959	37.5/0.961	37.7/0.961	37.4/0.960	37.5/0.961	**38.2**/**0.963**
	Average	33.7/0.921	35.4/**0.931**	34.6/0.928	35.2/**0.931**	35.2/0.930	34.9/0.928	35.1/0.930	**35.8**/**0.931**
x4	Img1	30.9/0.890	32.1/0.902	31.6/0.894	31.9/0.899	31.9/0.899	31.8/0.899	31.9/0.901	**32.2**/**0.909**
	Img2	33.8/0.953	35.8/0.961	34.7/0.955	35.5/0.960	35.4/0.959	35.1/0.956	35.3/0.959	**36.6**/**0.977**
	Img3	29.3/0.776	30.2/0.792	30.0/**0.795**	**30.2**/0.794	30.1/0.790	30.0/0.789	30.1/0.794	**30.2**/0.789
	Img4	31.6/0.933	33.2/**0.939**	32.4/0.935	33.0/0.938	32.9/0.936	32.8/0.936	32.9/0.938	**33.7**/0.927
	Img5	33.6/0.945	35.8/**0.952**	34.6/0.947	35.4/0.951	35.4/0.949	35.2/0.949	35.3/0.950	**36.3**/**0.952**
	Average	31.9/0.899	33.4/0.909	32.7/0.905	33.2/0.909	33.2/0.907	33.0/0.906	33.1/0.908	**33.8**/**0.911**

these methods is the reconstruction time from a pre-trained dictionary. The SRGAN is conducted on a Ubuntu server with RTX 2080Ti by Tensorflow, SRCNN is conducted using MatLab without GPU acceleration, we present their forward inference time in this table. The running time of bicubic interpolation is too short that can be neglected, therefore we did not show it.

As can be seen from Table 5, SRGAN is the fastest method because of GPU parallel acceleration, and x4 scale leads to longer inference time since it has one more upsample layer compared to x2 and x3. The SRCNN without GPU acceleration runs relatively slow, and the inference time is irrelevant to scale factor because it requires the input image has to be pre-interpolated to the target size, which means the computation complexity remains unchanged for different scales. The running time of all the dictionary-based methods decreases with the scale factor increasing because a small scale factor leads to a larger input size, which indicates more local patches need to be processed. Among those dictionary methods, NE + NNLS is the slowest one, followed by NE + LLE and NE + LS, which all involve neighbor embedding operation. Our method is not the fastest one but remains unchanged with different scale factors.

5. Conclusion

This paper introduces an improved single image SR algorithm. The key point of this method is to separate the data-fitting term and regularization term by introducing an auxiliary variable to replace the



(a) Bicubic (b) Zeyde (c) GR (d) ANR

(e) NE+LS (f) NE+NNLS (g) NE+LLE (h) Proposed

Fig. 5. SR results of Img2 with scale x2.

Table 5
Average running time of different algorithms on different datasets (unit: second).

Dataset	Scale	Zeyde	GR	ANR	NE + LS	NE + NNLS	NE + LLE	SRGAN (GPU)	SRCNN (CPU)	Proposed
Set5	x2	2.62	0.40	0.54	3.06	20.86	3.44	0.02	5.08	1.23
	x3	1.22	0.26	0.34	1.41	8.07	1.52	0.02	5.10	1.24
	x4	0.70	0.20	0.25	0.76	4.62	0.89	0.06	5.02	1.26
Set14	x2	5.10	0.80	1.07	5.78	40.07	6.58	0.02	10.69	3.89
	x3	2.39	0.51	0.66	2.66	16.66	2.98	0.02	10.62	3.80
	x4	1.54	0.42	0.51	1.61	8.96	1.88	0.04	11.42	3.81
B100	x2	3.74	0.58	0.78	4.07	26.93	4.55	0.07	7.23	2.38
	x3	1.63	0.34	0.44	1.79	11.02	2.03	0.07	7.02	2.13
	x4	0.99	0.27	0.34	1.06	5.95	1.23	0.09	7.16	2.39

estimation of SR result in the regularization term. With which the optimization can be carried out through a two-phase alternative renewal scheme. We also present a GIF denoiser to update the auxiliary variable. The experimental results indicate our method has a better performance than several classical model-based methods and deep-learning methods in terms of PSNR, but can not compete with deep learning-based method when comes to SSIM, we will continue on this work to improve it.

Funding

National Natural Science Foundation of China (Nos 51801142), Natural Science Foundation of Shaanxi Province (Nos 2018JQ5022), China Scholarship Council (201806960036), and Fundamental Research Funds for the Central Universities (JB180502).

Declaration of Competing Interest

We declare that there is no conflict of interest in this paper!

References

[1] Sky McKinley, Megan Levine, Cubic spline interpolation, College Redwoods 45 (1) (1998) 1049–1060.
[2] Jan Allebach, Ping Wah Wong, Edge-directed interpolation, in: Proceedings of 3rd IEEE International Conference on Image Processing, vol. 3, IEEE, 1996.
[3] Li, Yanghao, et al. Scale-aware trident networks for object detection, arXiv preprint arXiv:1901.01892, 2019.
[4] Jiahui Yu, et al., Generative image inpainting with contextual attention, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2018).
[5] Jiwon Kim, Jung Kwon Lee, Kyoung Mu Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[6] David L. Donoho, Michael Elad, Proceedings of the National Academy of Sciences, 2003, pp. 2197–2202.
[7] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, Yi Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 171–184, https://doi.org/10.1109/TPAMI.2012.88.
[8] David L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.
[9] Jianchao Yang, et al., Image Super-resolution as Sparse Representation of Raw Image Patches, IEEE, 2008.
[10] Roman Zeyde, Michael Elad, Matan Protter, On single image scale-up using sparse-representations, International Conference on Curves and Surfaces, Springer, Berlin, Heidelberg, 2010.
[11] Jianchao Yang, et al., Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.
[12] Tomer Peleg, Michael Elad, A statistical prediction model based on sparse representations for single image super-resolution, IEEE Trans. Image Process. 23 (6) (2014) 2569–2582.
[13] Weisheng Dong, et al., Nonlocally centralized sparse representation for image restoration, IEEE Trans. Image Process. 22 (4) (2012) 1620–1630.
[14] Chih-Yuan Yang, Jia-Bin Huang, Ming-Hsuan Yang, Exploiting self-similarities for single frame super-resolution, Asian Conference on Computer Vision, Springer, Berlin, Heidelberg, 2010.
[15] Jiaying Liu, et al., Retrieval compensated group structured sparsity for image super-resolution, IEEE Trans. Multimedia 19 (2) (2016) 302–316.
[16] Jiahe Shi, Chun Qi, Low-rank sparse representation for single image super-resolution via self-similarity learning, 2016 IEEE International Conference on Image Processing (ICIP), IEEE, 2016.
[17] Xiao-Yuan Jing, et al., Super-resolution person re-identification with semi-coupled low-rank discriminant dictionary learning, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2015).
[18] Xiaoxuan Chen, Chun Qi, Low-rank neighbor embedding for single image super-resolution, IEEE Signal Process Lett. 21 (1) (2013) 79–82.
[19] Ran He, Wei-Shi Zheng, Tieniu Tan, Zhenan Sun, Half-quadratic-based iterative minimization for robust sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 261–275, https://doi.org/10.1109/TPAMI.2013.102.
[20] Kaiming He, Jian Sun, Xiaoou Tang, Guided image filtering, IEEE Trans. Pattern Anal. Mach. Intell. 35 (6) (2012) 1397–1409.

[21] Zhengguo Li, et al., Weighted guided image filtering, IEEE Trans. Image Process. 24 (1) (2014) 120–129.
[22] Shutao Li, Xudong Kang, Hu. Jianwen, Image fusion with guided filtering, IEEE Trans. Image Process. 22 (7) (2013) 2864–2875.
[23] Cuong Cao Pham, Synh Viet Uyen Ha, Jae Wook Jeon, Adaptive guided image filtering for sharpness enhancement and noise reduction, Pacific-Rim Symposium on Image and Video Technology, Springer, Berlin, Heidelberg, 2011.
[24] Chih-Yuan Yang, Ming-Hsuan Yang, Fast direct super-resolution by simple functions, Proceedings of the IEEE International Conference on Computer Vision, (2013).
[25] Ra.du. Timofte, Vincent De Smet, Luc Van Gool, Anchored neighborhood regression for fast example-based super-resolution, Proceedings of the IEEE International Conference on Computer Vision, (2013).
[26] Marco Bevilacqua et al. Low-complexity single-image super-resolution based on nonnegative neighbor embedding, 2012, pp. 135-1.
[27] Hong Chang, Dit-Yan Yeung, Yimin Xiong, Super-resolution through neighbor embedding, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol. 1, IEEE, 2004.
[28] Christian Ledig, et al., Photo-realistic single image super-resolution using a generative adversarial network, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2017).
[29] S. Lei, Z. Shi, Z. Zou, Super-resolution for remote sensing images via local–global combined network[J], IEEE Geosci. Remote Sens. Lett. 14 (8) (2017) 1243–1247.
[30] Juntao Guan, Rui Lai, Ai Xiong, Zesheng Liu, Lin Gu, Fixed pattern noise reduction for infrared images based on cascade residual attention CNN, Neurocomputing 377 (2020) 301–313, https://doi.org/10.1016/j.neucom.2019.10.054.
[31] Rui Lai, Yongxue Li, Juntao Guan, Ai Xiong, Multi-scale visual attention deep convolutional neural network for multi-focus image fusion, IEEE Access 7 (2019) 114385–114399.
[32] Chao Dong, et al., Image super-resolution using deep convolutional networks, IEEE Trans Pattern Anal Mach Intell 38 (2) (2014).