GUN: Gradual Upsampling Network for single image super-resolution

Yang Zhao, *Member*, *IEEE*, Ronggang Wang, *Member*, *IEEE*, Weisheng Dong, *Member*, *IEEE*, Wei Jia, *Member*, *IEEE*, Jianchao Yang, Xiaoping Liu, *and* Wen Gao, *Fellow*, *IEEE*

***Abstract*—In this paper, we propose an efficient super-resolution (SR) method based on deep convolutional neural network (CNN), namely gradual upsampling network (GUN). Recent CNN based SR methods either preliminarily magnify the low resolution (LR) input to high resolution (HR) and then reconstruct the HR input, or directly reconstruct the LR input and then recover the HR result at the last layer. The proposed GUN utilizes a gradual process instead of these two kinds of frameworks. The GUN consists of an input layer, multistep upsampling and convolutional layers, and an output layer. By means of the gradual process, the proposed network can simplify the difficult direct SR problem to multistep easier upsampling tasks with very small magnification factor in each step. Furthermore, a gradual training strategy is presented for the GUN. In the proposed training process, an initial network can be easily trained with edge-like samples, and then the weights are gradually tuned with more complex samples. The GUN can recover fine and vivid results, and is easy to be trained. The experimental results on several image sets demonstrate the effectiveness of the proposed network.**

***Index Terms*—Super-resolution, upsampling, deep convolutional neural network**

I. INTRODUCTION

SINGLE image super-resolution (SISR), which is also known as image upsampling, image upscaling, or magnification, is a classical problem in computer vision and image processing. Generally, the aim of SISR is to reconstruct a high-quality (HQ) and high-resolution (HR) image from a single low-resolution (LR) input. It is a typical ill-posed problem since lots of detailed information is lost during the downsampling process. How to recover a HQ and HR image with low cost is still a fundamental and challenging task.

The basic SISR method is interpolation-based algorithm, such as nearest-neighbor, linear, bilinear, bicubic [1-2]. Unfortunately, interpolating process often causes blurring, jaggy, and ringing effects. Hence, many methods have been proposed to suppress these unnatural artifacts by means of different strategies, such as introducing edge prior knowledge [3-5], altering interpolated grid [6-9], and sharpening the edges [10-12], etc. These improved methods refine the unnatural artifacts, but they still cannot recover extra details.

Reconstruction-based algorithm is another type of classical SISR method. This kind of method is based on a fundamental constraint that the reconstructed HR image should be consistent with the original LR input. In order to reproduce shaper and clearer image, many extra constraints or image models have been proposed over the years, e.g., gradient-based constraints [13-18], local texture constraint [19], total variation regularizer [20-21], deblurring-based models [22-24], etc. However, the performance of these reconstruction-based algorithms often degrades rapidly when the magnification factor is large, because the basic similarity constraint is defined on the LR space.

Example-based or learning-based method has received increasing attention in recent years. This kind of algorithm tries to reconstruct the missing details via lots of known LR/HR example-pairs. Learning-based method is first presented in [25] and further developed in [26-50]. Many typical and effective learning-based models have been proposed, such as neighbor embedding based algorithms [26-29], sparse representation based methods [30-37], and local self-exemplar models [38-41]. Although these methods can recover sharp edges with fine details, the computation cost of them is quite high. The most time-consuming process of these methods is the patch-by-patch optimization of representation coefficients or weights. Recently, some fast and high-performance SISR models have been presented, i.e., anchor neighborhood regression methods [42-43], and SISR forests [59, 66]. These two kinds of models can obtain obvious speedup by means of pre-computing the projection matrix offline or utilizing the efficient random forest algorithm.

In recent three years, deep neural network (DNN) based methods have been widely applied in computer vision tasks and have achieved impressive results. Many DNN based SISR algorithms have also been proposed. Dong *et al.* [44] presented an effective SISR method by means of shallow convolutional neural network (CNN). Kim *et al.* [45] further improve this method by successfully training a narrower and deeper CNN for SISR. Furthermore, many other DNN models have been applied in super-resolution (SR) scenario, such as deep residual

This work was partly supported by the grant of National Science Foundation of China 61402018, 61370115, 61305093, china 863 project of 2015AA015905, Shenzhen Peacock Plan JCYJ20150331100658943, and Guangdong Province Project 2014B010117007 for funding.

Y. Zhao, W. Jia and X. Liu are with the School of Computer and Information, Hefei University of Technology, 193 Tunxi Road, Hefei 230011, China (email: yzhao@hfut.edu.cn, icg.jiawei@gmail.com, lxp@hfut.edu.cn)

Y. Zhao, R. Wang, and W. Gao are with the School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, 2199 Lishui Road, Shenzhen 518055, China (email: zhaoyang@pkusz.edu.cn; rgwang@pkusz.edu.cn; wgao@pku.edu.cn)

W. Dong are with School of Electronic Engineering Xidian University, Xi'an, China. (email: wsdong@mail.xidian.edu.cn)

J. Yang is with the Snap Inc., Venice, CA 90291, USA (email: jianchao.yang@snapchat.com)

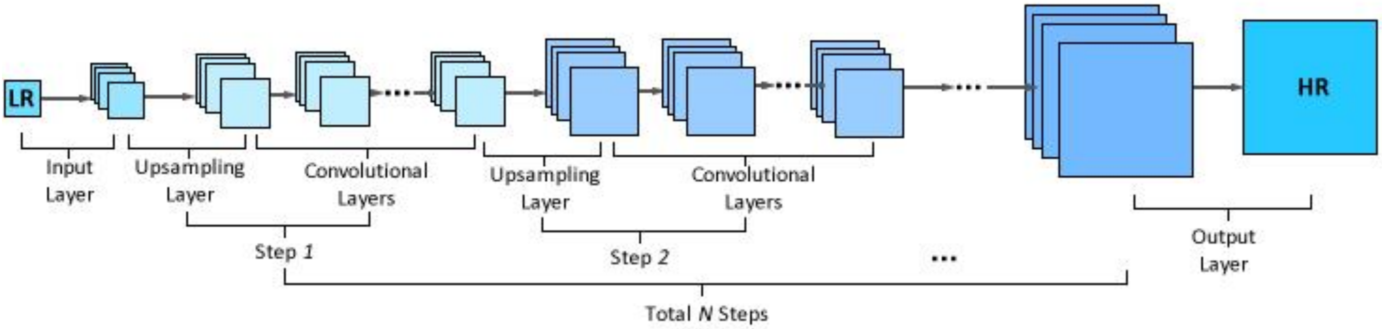


Fig. 1 The framework of the proposed GUN

network [55], sparse convolutional network [48, 60], combined deep and shallow CNN [47], mixed CNN [51], recursive convolutional network [46], recurrent residual network [52], bidirectional recurrent convolutional network [54], collaborative local auto-encoder [53] and so on. In the CNN based methods, the SISR is treated as an image reconstruction problem, and a general convolutional network without pooling and fully-connected layers is often used. The CNN has a strong ability to fit a highly nonlinear regression problem, and these CNN based SR methods thus achieve state-of-the-art results. The upsampling strategies in these CNN based methods can be roughly divided into two categories: some methods preliminarily magnify the LR input to high resolution and then utilize the network to reconstruct the HR inputs [44-48]; some other methods directly reconstruct the LR input by means of convolutional networks and then reform the HR result in the last layer [49-50]. Note that another upsampling strategy, gradually magnifying, has not been studied in these CNN based methods. Most recently, the generative adversarial network (GAN) based SR methods [55, 56] can reproduce amazing texture, and the GAN also reveals a possible way to recover the missing textural details.

In this paper, we propose a gradual upsampling network (GUN), which introduces several upsampling layers and gradually reconstructs the LR input to larger resolution step-by-step. Different to the methods [44-45], which directly obtain an initial HR input and then reconstructing the HR input via deep network, the proposed GUN can be regarded as the concatenation of many sub-networks. In each sub-network, the target magnification factor is very small and thus the difficulty of training in each step can be reduced. Gradual upsampling is a commonly used strategy in SR and other similar ill-posed problems. For example, direct magnifying with large factor is difficult in local self-exemplar methods, because merely finite self-examples can be used. Some methods [39] thus utilize the multistep magnification and adopt a very small factor in each step to reduce the difficulty of reconstruction. To take another example, it is also very hard to generate a HR image in generative model, the Laplacian pyramid GAN [57] and a CNN-based method [58] both enforce gradual upscaling process to yield a final full resolution output. Compared to the state-of-the-art networks [44-45], the proposed GUN has the following merits,

1) First, the gradual upsampling can relax the difficult direct magnification task to several easier upsampling problems with very small factor. The GUN is thus easier to be trained.
2) Second, a gradual training method is presented for the GUN. The gradual training process can rapidly train an initial network with finite simple samples. The GUN is then gradually optimized to reproduce HQ and HR results by continually adding more complicated training samples to the training process.
3) Moreover, comparing with some direct upsampling methods [44-48], the gradual upsampling can reduce the resolution of feature maps during the convolutional process, and therefore can decrease the computational cost.
4) Experimental results on several datasets demonstrate that the proposed GUN can recover clear and natural images and outperform some state-of-the-art methods.

The following paragraphs of this paper are organized as follows. Section II introduces the proposed GUN and gradual training process in details. Section III presents some implementation details. Experimental results are given in Section IV, and Section V concludes the paper.

II. Gradual Upsampling Network

A. *The proposed network*

As illustrated in Fig.1, the proposed GUN consists of several layers, i.e., the input layer, the multistep upsampling and convolutional layers, and the final output layer. In order to concisely illustrate the network, the activation and batch normalization (BN) layers are not shown in Fig.1. We describe each component of the GUN in the following.

1) *Input layer*

Similar to the CNN based SR methods [44, 45], the input layer of GUN is also a typical convolutional (conv.) layer activated by the rectified linear units (ReLU). Hence, given a LR input $\boldsymbol{y}$ of size (m_L, n_L), the output of the input layer is,

$$F_{in}(\boldsymbol{y}) = ReLU(\boldsymbol{\omega}_{in} * \boldsymbol{y} + \boldsymbol{b}_{in})$$

In this paper, we use 64 filters with the size of 3×3 in the first layer, and thus the sizes of weights $\boldsymbol{\omega}_{in}$ and bias term $\boldsymbol{b}_{in}$ are $3 \times 3 \times 1 \times 64$ and 1×64, respectively. It should be noted that all the conv. processes in the GUN are with zero-padding, so that the resolution is invariant after the conv. process.

2) *Upsampling layer*

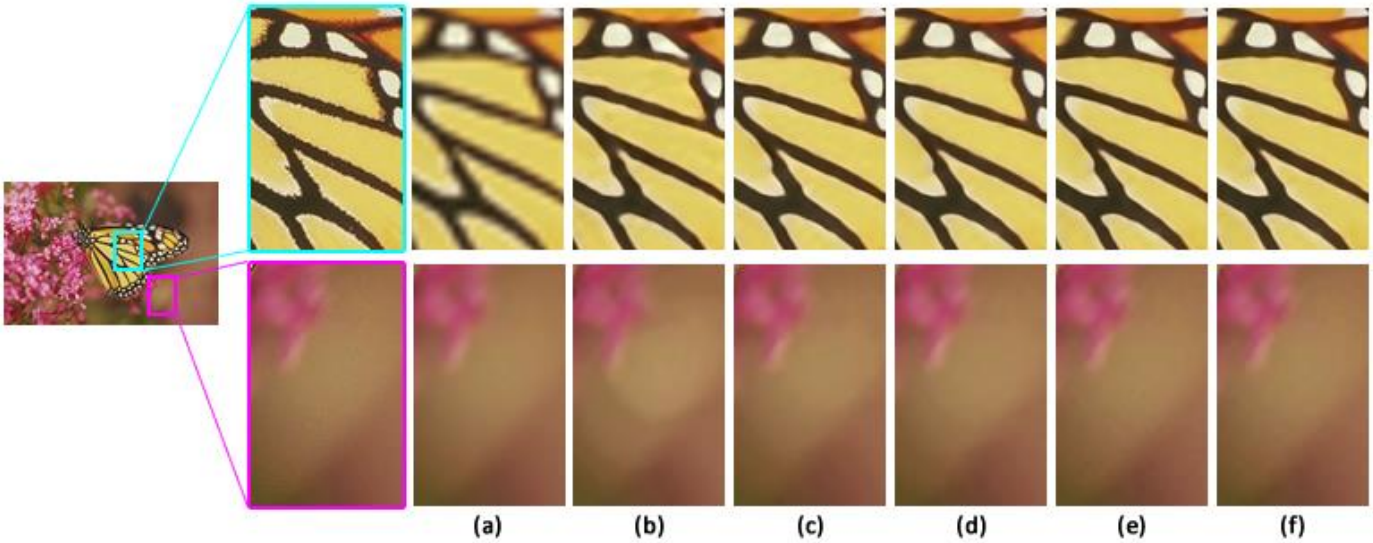


Fig. 2 The 4 × upsampling results of different training stages, (a) Bicubic results (b) training with $\lambda = 1.2$, (c) training with $\lambda = 1$, (d) training with $\lambda = 0.8$, (e) training with $\lambda = 0.5$, (f) training with $\lambda = 0$.

The upsampling layer can resize the input to a slightly larger resolution, which can be described as,

$$F_{up}^{l}(\boldsymbol{y}^{l}) = U\uparrow(\boldsymbol{y}^{l})$$

where $\boldsymbol{y}^{l}$ is the input of the layer l, and $U\uparrow$ denotes the upsampling process. In this paper, the traditional bicubic interpolation is adopted in the upsampling layers. Correspondingly, the propagated error value passed by this layer need to be downsampled during the back propagation process:

$$\boldsymbol{\delta}^{l} = D\downarrow(\boldsymbol{\delta}^{l+1}).$$

where δ^{l} denotes error value of the l-th layer, and $D\downarrow$ represents the downsampling process.

Note that the de-convolutional (de-conv.) or un-pooling layer can also enlarge the resolution of the output. However, the un-pooling layer can merely upscale with a fixed integer factor, and the enlarged size of de-conv. layer is strictly determined by the size of filters or the stride of de-conv. process. Comparing to un-pooling layer and de-conv. layer, the upsampling layer can magnify the input to a specified resolution more freely[1].

3) *Convolutional layers*

In the proposed GUN, the LR input is gradually upsampled within many steps. Each step contains an upsampling layer and several conv. layers. Motivated by the VGG-Net [61] and VDSR [45], the stack of many 3 × 3 conv. layers can have an effective receptive field of larger size, e.g., 5 × 5, 7 × 7, and so on. Therefore, we also set the size of filtersto 3 × 3. However, the multiple 3 × 3 convolutions cannot represent 1 × 1 convolution. It has been proven that 1 × 1 convolution is valid in the SR problem [44]. We thus add a 1 × 1 conv. layer as the last layer in each step. Then the output of the conv. layer l can be calculated as,

$$F^{l}(\boldsymbol{y}^{l}) = ReLU(\boldsymbol{\omega}^{l} * \boldsymbol{y}^{l} + \boldsymbol{b}^{l})$$

We also utilize 64 feature maps in all the conv. layers. Hence, the size of $\boldsymbol{\omega}^{l}$ is $1 \times 1 \times 64 \times 64$ for the last conv. layer in each step, and the size of $\boldsymbol{\omega}^{l}$ for all the other conv. layer is $3 \times 3 \times 64 \times 64$. The size of $\boldsymbol{b}^{l}$ is therefore 1×64 for all the conv. layers. In addition, a BN layer is utilized after each conv. layer in the GUN to enhance the capacity of the network.

4) *Output layer*

The final output layer is computed as,

$$\boldsymbol{x} = \boldsymbol{\omega}_{out} * \boldsymbol{y}^{l} + \boldsymbol{b}_{out}$$

where the sizes of $\boldsymbol{\omega}_{out}$ and $\boldsymbol{b}_{out}$ are $3 \times 3 \times 64 \times 1$ and 1×1, respectively. The final output is HR image $\boldsymbol{x}$ with the size of (m_H, n_H).

Suppose the LR input $\boldsymbol{y}$ is gradually reconstructed with total N steps, the upsampled resolution of the i-th step is,

$$(m_L + i\Delta_m,\ n_L + i\Delta_n),\quad i = 1, 2, \cdots, N-1,$$

where,

$$\Delta_m = R\left(\frac{m_H - m_L}{N}\right),$$

$$\Delta_n = R\left(\frac{n_H - n_L}{N}\right),$$

$R(\cdot)$ denotes the round down function. Note that the upsampled resolution of the N-th step is fixed to (m_H, n_H), so that the resolution of the final output is the same with the target.

[1] A gradually de-conv. network is also presented similar to the GUN. More details and related experimental results can be found in the supplementary materials: http://yzhaocv.weebly.com/projectpage/gun-supplementary

B. Training of the GUN

1) Normal training settings

As in other SR networks, the average mean squared error (MSE) is also used as the loss function for the GUN. The MSE loss can restrict that the pixel-wise contents of the output are exactly consistent with that of the HR sample. Given a training image set $\{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n=1}^{N_s}$, the GUN can be trained by minimizing the following MSE loss function,

$$L(\Theta) = \frac{1}{N_s}\sum_{n=1}^{N_s} \|f(\boldsymbol{y}^n;\Theta) - \boldsymbol{x}^n\|^2$$

where $f(\cdot)$ denotes the output of the network, and Θ is the weight set of the GUN.

In this paper, we adopt the training image set proposed by Yang *et al.* [30], which is also used in many other learning-based methods [30-35, 42-45]. This training set contains 91 images downloaded from the internet. Many sample patches are randomly selected from each training image with overlapping, and these patches are further augmented by rotating with three orientations (45 °, 90 °, and 180 °).

The proposed GUN is trained by utilizing mini-batch gradient descent based on backward propagation. Each mini-batch contains 64 image patches. The momentum parameter and weight decay are set as in [45]. The learning rate is initially set to 10^{-4} and then decreased by a factor of 10 after every 5 epochs. We implement the GUN [2] by means of the MatConvNet[3] package [62].

2) Gradual Training: From easy to difficult

Image set is very important to train an effective deep network. Related works often enlarge the quantity of training samples to refine the performance, e.g., adding more images into the training set, extracting more patches from one image, and data augmentation (flipping, rotation, and so on). However, all these samples play the same role during the training process. In traditional learning-based SR methods, it can be found that the edge-like patterns with stable local structure are much easier to be learned in the dictionary than other kinds of patches. For example, most of the dictionary atoms learned via sparse representation or clustering are edge-like local patterns. Is it also easier to train the network by means of these edge-like patches? To answer this question, we adopt a gradual training process. In the gradual training, the patches which contain sharp edges are firstly chosen as the initial training set. Patches with flatter structure are then gradually added into the training set. By using the proposed training process, the GUN is firstly trained to magnify the sharp edges, and the details of sharp edge area are relatively easier to be learned. The network then learns to reconstruct more difficult situations by gradually fine-tuning the weights with more training samples.

TABLE I. AVERAGE PSNR (DB) ON 'SET14' WITH DIFFERENT GRADUAL TRAINING STAGES

λ	2 ×	3 ×	4 ×
Bicubic	30.36	27.67	26.12
1.2	32.77	29.45	27.62
1	33.06	29.67	27.89
0.8	33.24	29.96	28.15
0.5	33.30	30.04	28.21
0	33.33	30.07	28.25

In this paper, the edge-like patches are selected by means of the average local gray value difference (ALGD), which can be computed as,

$$\mathrm{v}_{ALGD} = \sum_{p=1}^{N_p}(g_p - \bar{g})$$

where g_p $(p = 1,2,\cdots,N_p)$ denotes a pixel in an image patch, N_p is the total number of pixels in that patch, and $\bar{g}$ denotes the average gray value of the patch. The edge-like patches can then be selected by comparing the ALGD value with the average ALGD value of the whole training set ($\overline{\mathrm{v}_{ALGD}}$) as follows,

$$\mathrm{v}_{ALGD} \geq \lambda\overline{\mathrm{v}_{ALGD}}$$

where λ is an artificial parameters, and the λ is orderly set as *1.2, 1, 0.8, 0.5,* and *0* in our training process. At the first, the sharp-edge-patches with $(\lambda = 1.2)$ are utilized to form the initial training set. Take 4 × magnification for example, merely about 29, 000 12 × 12 patches are extracted. This training stage is convergence very fast within first 3-5 epochs, and it is much faster than the normal training process since the size of the initial training set is much smaller. The reconstructed results of different training stages are illustrated in Fig.2. From Fig.2 (b) we can find that the GUN can learn to magnify sharp edges with faster training and less samples. But unfortunately, the flat area is also over-sharpened. We then feed more patches with $(\lambda = 1)$ to the network, and the number of samples is increased to almost 82, 000 for 4 × magnification. This training stage costs another 3 epochs. The upsampling results after this stage is shown in Fig.2(c). It can be found that the reconstructed edges become clearer, and the flat area is much better than the former result. Similarly, the patches with $(\lambda = 0.8, 0.5, \text{and } 0)$ are added into the training set in turn, and finally over 300, 000 samples are utilized. By comparing the results in Fig.2, we can find that the gradual training process can reproduce both sharp edges and natural flat area by gradually tuning the weights. We implement each training stage about 3 epochs, and totally use 15-20 epochs to train the GUN. This proposed training process roughly cost several hours on a personal computer using a Titan X GPU.

The average PSNR values of 2 × , 3 × , and 4 × magnifications on image set 'Set14' of different training stages are listed in the Table I. From which we can find that the initial training with small training set can already obtain fine PSNR results. These results are then improved by tuning the network with more and complex training samples. Furthermore, the results of the gradual training also verify the prior knowledge

[2] The demo codes of the GUN can be downloaded from the following website: http://yzhaocv.weebly.com/projectpage/gun

[3] http://www.vlfeat.org/matconvnet/

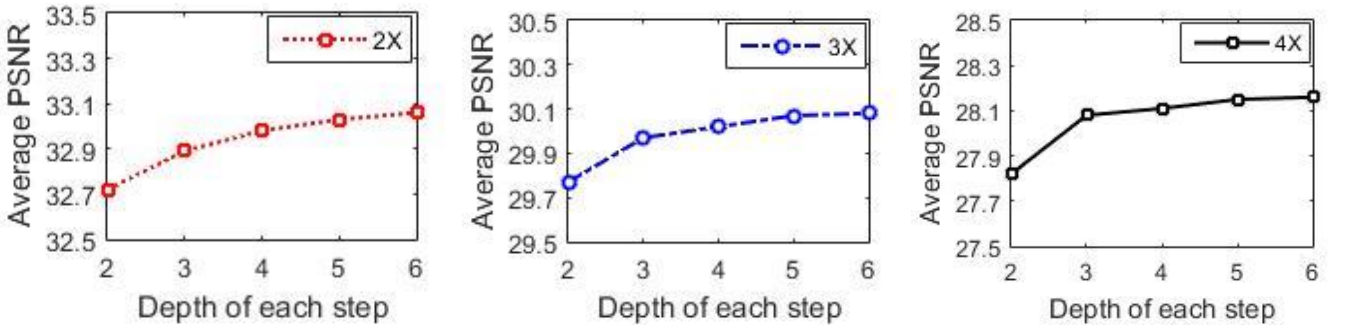
about the training of SR network, i.e., the reconstruction of sharp edges with stable local structure are easier to be learned, and gradual training can make the network learn the SR task better, from easy to difficult.

C. Discussions of the computation complexity

The computation complexity of the GUN can be computed as,

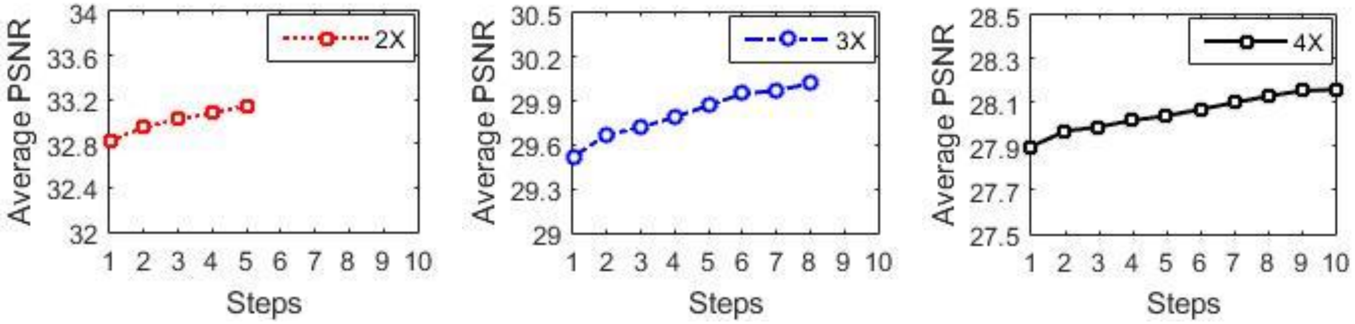
$$O\{(f_1^2 q_1)S_1 + \sum_{i=1}^{N}\left(\sum_{l=1}^{D} p_l f_l^2 q_l\right)S_i + (p_L f_L^2)S_L\}$$

where S_1, S_i, and $S_L$denote the size of the LR image, the size of the i-th step maps, and the size of the HR output, respectively. The f_l is the filter size of the l-th layer, p_l/q_l represent the number of input/output feature maps of the l-th layer, and D denotes the depth of the sub-network in each step. It can be observed that the complexity is proportional to the size of image, the number of feature maps in each layer, filter size, and the depth of the network. As mentioned before, the p_l/q_l is fixed as 64, and the filter size is set as small as 3 or 1 in the proposed network to avoid high computation complexity. Furthermore, the GUN has lower computation complexity than the direct upsampling network [44-45], since the S_i are smaller than the S_L.



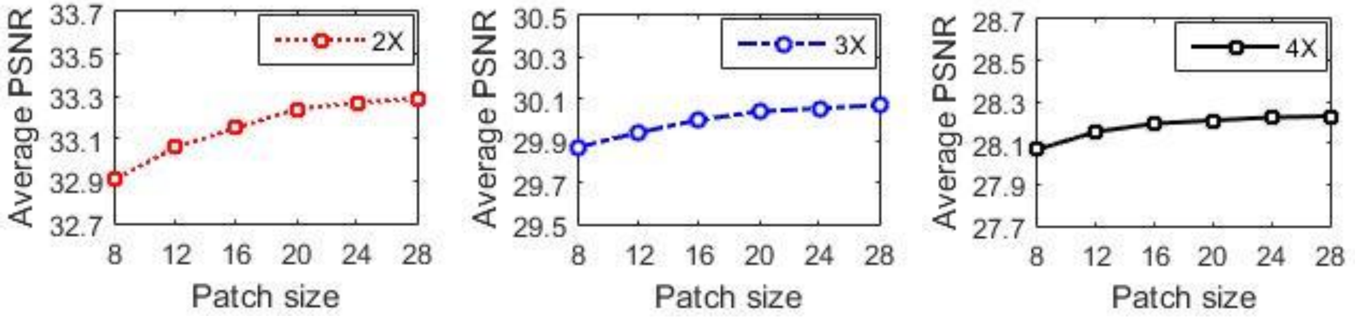
(a) (b) (c)

Fig.3 Average PSNR values with different depths in each step on 'Set14', (a) 2 × magnification, (b) 3 × magnification, (c) 4 × magnification.



(a) (b) (c)

Fig.4 Average PSNR values with different number of steps on 'Set14', (a) 2 × magnification, (b) 3 × magnification, (c) 4 × magnification.



(a) (b) (c)

Fig.5 Average PSNR values with different patch sizes on 'Set14', (a) 2 × magnification, (b) 3 × magnification, (c) 4 × magnification.

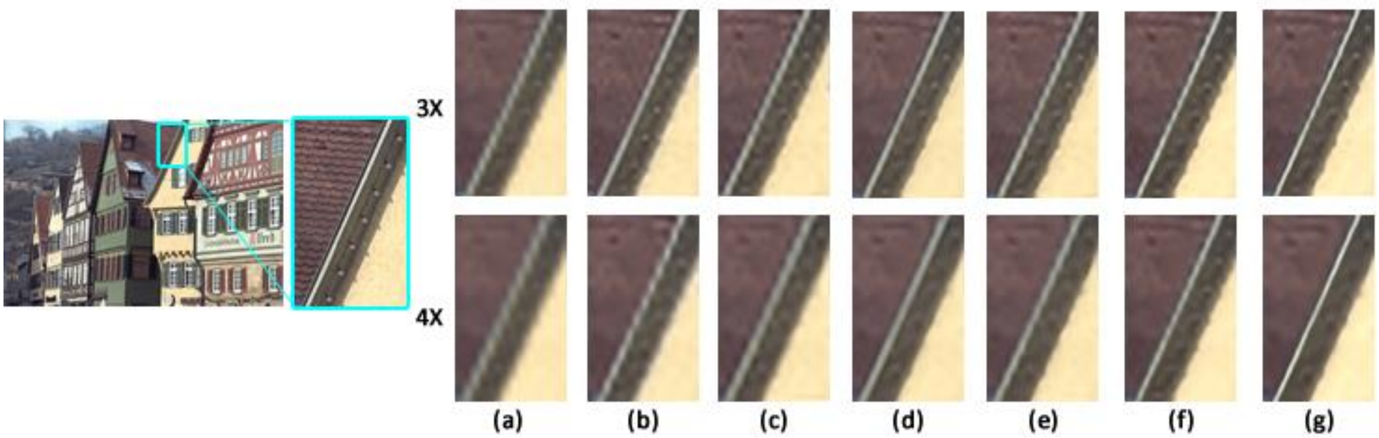


Fig.6 3 × and 4 × upsampled results of 'kodim08' image with different methods, (a) with bicubic, (b) with the ASDS [32], (c) with the ANR [42], (d) with the A+ [43], (e) with the SRCNN [44]. (f) with the VDSR [45], (g) with the GUN.

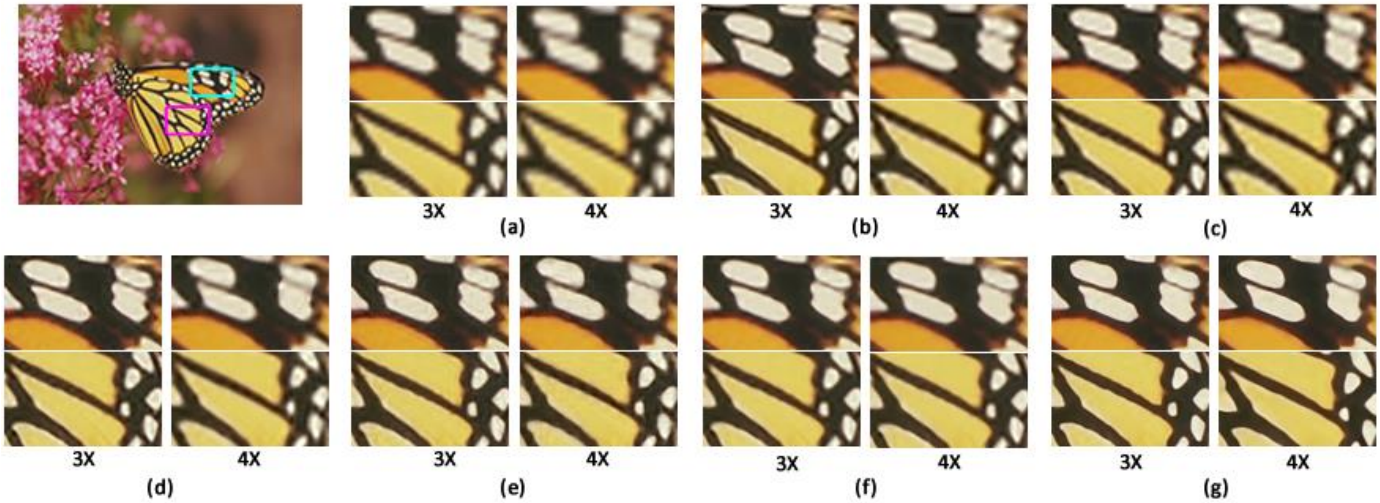


Fig.7 3 × and 4 × upsampled results of 'monarch' image with different methods, (a) with bicubic, (b) with the ASDS [32], (c) with the ANR [42], (d) with the A+ [43], (e) with the SRCNN [44]. (f) with the VDSR [45], (g) with the GUN.

III. Implementation Details

A. Selection of the depth

In [45], Kim *et al.* have proven that the deeper network can obtain better SR results. Hence, it is important to select an appropriate depth of the network. In the GUN, the depth of the network is mainly decided by two parameters, i.e., the number of upsampling steps, and the depth of the sub-network in each step. In the following, we test the selection of these two parameters separately.

1) Selection of depth in each step

Fig.3 illustrates the average PSNR values on 'Set14' with different depths in each step. The total number of steps is fixed as 4, and the size of input patch is 12 × 12. Each step consists of several 3 × 3 conv. Layers and one last 1 × 1 conv. layer. As shown in Fig.3, it can be found that the deeper network also performs the better. However, the PSNR results increase slowly when the depth is larger than 4, presumably because of the fixed training settings and finite training samples. The deeper network also costs more training and testing time. We thus set the depth in each step to 4 in the following experiment.

2) Selection of steps

Fig.4 shows the relationship between the average PSNR results and the total number of steps. The patch size is also 12 × 12, and the depth in each step is fixed to 4. On the whole, increasing the number of steps can improve the performance. This also demonstrates the deeper the network the better the results. Note that the number of steps is also influenced by the increased resolution between the LR and HR training patches. In this paper, we experimentally set the number of steps as 5, 8, and 9 for 2 ×, 3 ×, and 4 × magnification, respectively.

B. Size of the input patch

The average PSNR values with different patch sizes are shown in Fig.5. We can find that enlarging the patch can also slightly increase the reconstruction results. It should be noticed that the quantity of samples are fixed as 200, 000 to train the network. In practice, larger patch size also leads to the reduction of total training samples. As a result, we select the patch size by considering both the performance and the total number of training samples. In our experiment, the patch sizes are chosen as 20, 16, and 12 for 2 ×, 3 ×, and 4 × magnification, respectively.

IV. Experimental Results

A. Testing image sets

For testing, we use three typical and largish image sets, i.e., ‘Set14’ [36], ‘B100’ [42], and Kodak PhotoCD dataset[4]. ‘Set14’ contains 14 commonly used images in many SR literatures. ‘B100’ selects 100 images from the Berkeley Segmentation dataset (BSD) [54]. The Kodak PhotoCD dataset consists of 24 lossless true color images without compression artifacts, and is used as a standard testing set for many image processing works [66].

B. Comparison with state-of-the-arts

In this section, we compare the proposed GUN with some state-of-the-art learning-based methods, such as the LLE [26], the ScSR [30], the ASDS [32], the ANR [42], the A+ [43], the SRCNN [44], and the VDSR [45]. In our experiment, the color testing images are firstly converted from RGB to YUV, and these SR methods are applied only on Y (intensity) component. The rest U and V channels are simply reconstructed with bicubic interpolation. In our experiment, the LR inputs are obtained by downsampling the original HR images with bicubic interpolation. Note that the ASDS firstly filtered and then downsampled the HR image to obtain the LR input, which is slightly different to other methods. The magnification factors in this paper are set as 2, 3, and 4.

Fig.6 compares the 3 × and 4 × magnified results with different SR methods. The zoom-up area is marked with blue rectangle, and the HR ground truth is also given. By comparing the upsampled edges of the roof, we can make the following observations. First, the texture on the roof are totally lost during the 3 × and 4 × magnifying for all these state-of-the-art methods. Second, these learning-based methods can recover much better lines than bicubic. Third, the two deep networks, i.e., the VDSR and the GUN, can reproduce sharp edges for 4 × SR. Last, by comparing the details around the roof edges, the proposed GUN obtains clearer and sharper results than other methods.

Fig.7 also illustrates the 3 × and 4 × upsampled results of ‘monarch’ image with different methods. We select two area on the wings of the monarch, from which we can get the following findings. First, the bicubic interpolated results are very jaggy and blurring, and the learning-based methods can reconstruct sharper edges. The results of the GUN are the sharpest among them. Second, the VDSR and the GUN can reproduce fine flat area than other methods, and that make their results much cleaner and clearer. Last, by comparing the tiny lines in the 4 × results, the GUN can recover more natural and better details by means of the gradual learning and upsampling.

Fig.8 compares the 3 × and 4 × SR results on another image. We can get some findings similar to Fig.6. The A+, the SRCNN, the VDSR and the proposed GUN can reconstruct clear and smooth necklace. By comparing the details of the necklace, the GUN still recovers sharper and clearer results than other state-of-the-art methods.

The 2 × results on image ‘PPT’ with various methods are shown in Fig.9. We can find these learning-based methods can recover fine edges for small magnification factor. To facilitate the comparison of subjective quality, the residual map between each result and the original HR image is illustrated. By comparing the details around the digits, we can obtain some observations. First, the result of Bicubic interpolation is blurry but without fake edges. Second, the VDSR and the GUN can reproduce clear edges, while other methods suffer from either ringing effects or fake edges. Last, by comparing the residual components, the GUN achieves the least difference to the HR ground truth.

For objective quality assessment, we utilize two common evaluation metrics of the PSNR and the SSIM [63]. However, these two indexes may not reflect image subjective quality very well. In [64], the information fidelity criterion (IFC) [65] is proved to have high correlation with human ratings for SR evaluation. Thus we also adopt the IFC in the experiment to estimate the subjective quality of different SR results. Table II, Table III, and Table IV list the objective assessment results on three datasets of the ‘Set14’, the ‘B100’, and the Kodak dataset, respectively. From these tables, we can find that the proposed GUN can achieve higher PSNR and SSIM values than other methods for different magnification factors and different datasets. By comparing the IFC values, the proposed GUN also obtains better IFC results than other methods. These results also demonstrate the effectiveness of the proposed network and the training processes.

C. Further analyses

Finally we again describe the benefits and further analyze the limitations of the proposed GUN. The gradual upsampling can reduce the difficulty of training a direct upsampling network. Furthermore, the proposed training process can rapidly train an initial network and then gradually optimize the weights. By comparing with the efficient deep network VDSR [45], the residual-training and gradient clipping strategies are not used in the proposed network, but the GUN is still converged very fast in the first several epochs. The GUN can be regarded as a variant of the VDSR which is specially designed for the SR scenario. But for other image reconstruction problems, such as denoising and deblocking, the VDSR is still much easier to be applied than the GUN, since the upsampling process maybe not needed in these tasks. Another obvious limitation of the GUN is that it still cannot recover fine textural details. As illustrated in Fig.8, the material of the pearl-necklace can be seen in the ground truth, but all the SR results lost these textural details. Note that the bicubic interpolated result is seemed not so smooth, that is because of the jaggy effect rather than recovering the true texture. It reveals that although recent methods, such as the VDSR and the proposed GUN, can recover fine HR image with vivid and sharp edges, how to accurately recover the missing textural components is still needed more attentions. Hence, we plan to combine with the creative adversarial model [55] to improve this limitation in the future work.

[4] http://r0k.us/graphics/kodak/

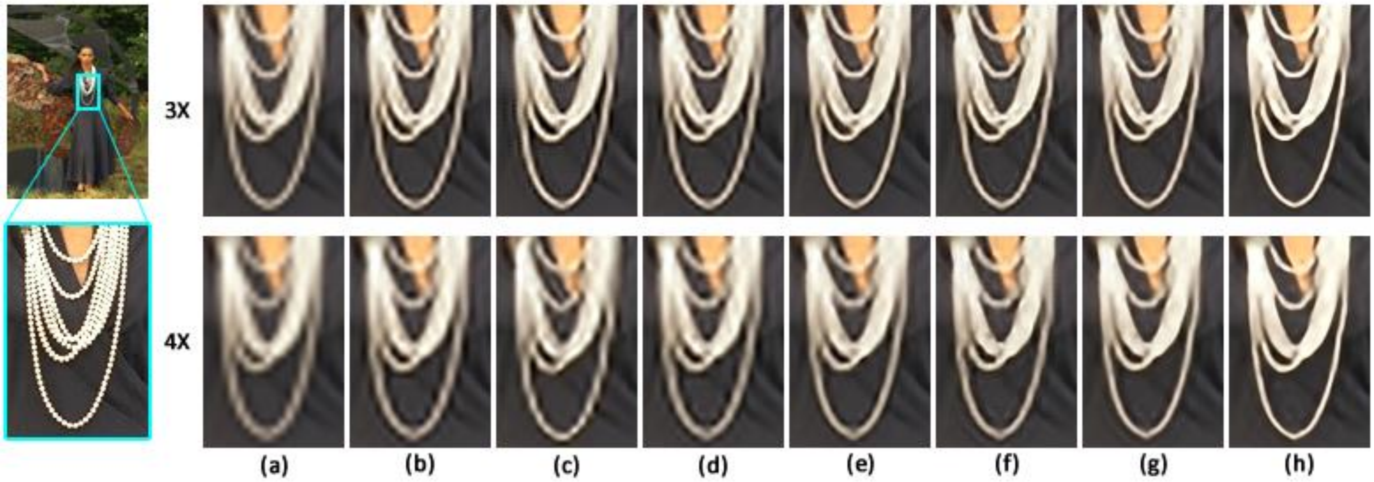


Fig.8 3 × and 4 × upsampled results of 'kodim18' image with different methods, (a) with bicubic, (b) with the LLE [26], (c) with the ASDS [32], (d) with the ANR [42], (e) with the A+ [43], (f) with the SRCNN [44]. (g) with the VDSR [45], (h) with the GUN.

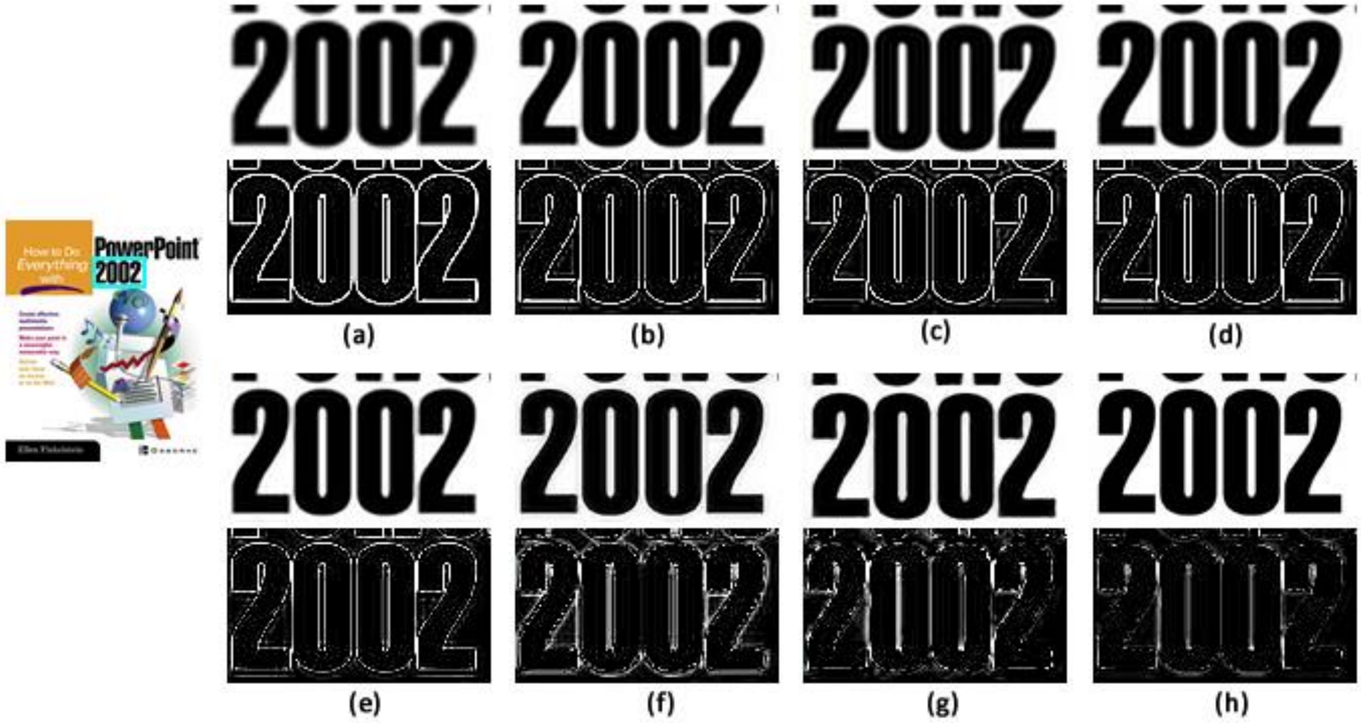


Fig.9 2 × SR results of 'PPT' image with different methods, (a) with bicubic, (b) with the LLE [26], (c) with the ASDS [32], (d) with the ANR [42], (e) with the A+ [43], (f) with the SRCNN [44]. (g) with the VDSR [45], (h) with the GUN. The residual map between each result and the original HR image is illustrated.

V.Conclusions

In this paper, we propose an efficient deep convolutional neural network based super-resolution method, namely gradual upsampling network (GUN). The proposed GUN consists of an input layer, multistep upsampling and convolutional layers, and an output layer. The difficult direct upsampling problem is relaxed to several easier gradual upsampling processes with very small magnification factor. Hence, the GUN can efficiently learn to reconstruct the HR results gradually. Furthermore, we present a gradual training process for the GUN, in which the simple edge-like patches are firstly utilized to train an initial network and then more complex patches are added to tuning the weights. Experimental results on three representative image datasets demonstrate that the proposed GUN can recover fine results and outperform some state-of-the-art methods.

Acknowledgements

The authors would like to sincerely thank the anonymous reviewers. We also sincerely thank R. Timofte, and C. Dong for sharing the source codes of the ANR/A+, and the SRCNN methods.

Table II.
Average PSNR (dB), SSIM, and IFC of different methods on image set “Set14”

	2X			3X			4X		
	PSNR	**SSIM**	**IFC**	**PSNR**	**SSIM**	**IFC**	**PSNR**	**SSIM**	**IFC**
Bicubic	30.36	0.9417	5.83	27.67	0.8596	3.41	26.12	0.7857	2.27
LLE	31.91	0.9587	6.08	28.74	0.8836	3.89	26.95	0.8137	2.21
ScSR	31.21	0.9620	6.22	28.01	0.8882	4.04	26.57	0.8183	2.65
ASDS	31.15	0.9627	6.61	27.91	0.8938	4.11	26.94	0.8190	2.35
Zeyde	31.96	0.9589	6.25	28.80	0.8841	4.02	26.99	0.8159	2.67
ANR	31.95	0.9626	6.36	28.80	0.8890	3.67	27.00	0.8194	2.48
A+	32.39	0.9641	6.54	29.12	0.8940	4.04	27.34	0.8294	2.62
SRCNN	32.93	0.9648	7.06	29.54	0.9023	3.94	27.85	0.8458	2.72
VDSR	33.07	0.9689	7.16	29.82	0.9055	4.17	28.07	0.8533	2.75
GUN	**33.35**	**0.9698**	**7.31**	**30.08**	**0.9112**	**4.31**	**28.29**	**0.8648**	**2.96**

Table III.
Average PSNR (dB), SSIM, and IFC of different methods on image set “B100”

	2X			3X			4X		
	PSNR	**SSIM**	**IFC**	**PSNR**	**SSIM**	**IFC**	**PSNR**	**SSIM**	**IFC**
Bicubic	29.35	0.8334	5.85	27.17	0.7361	3.47	25.95	0.6671	2.29
LLE	30.40	0.8674	6.12	27.84	0.7687	3.95	26.47	0.6937	2.74
ScSR	30.32	0.8709	6.24	27.74	0.7719	4.22	26.33	0.6997	2.85
ASDS	30.19	0.8712	6.72	27.65	0.7735	4.24	26.45	0.7003	2.97
Zeyde	30.40	0.8682	6.32	27.87	0.7693	4.18	26.51	0.6963	2.76
ANR	30.50	0.8706	6.59	27.90	0.7724	4.16	26.52	0.6991	2.67
A+	30.76	0.8762	6.64	28.18	0.7764	4.19	26.76	0.7062	2.72
SRCNN	31.06	0.8854	6.62	28.18	0.7780	4.14	26.79	0.7059	2.69
VDSR	31.30	0.8861	6.79	28.31	0.7789	4.24	26.93	0.7070	2.83
GUN	**31.49**	**0.8889**	**6.96**	**28.49**	**0.7886**	**4.32**	**27.15**	**0.7191**	**3.06**

Table IV.
Average PSNR (dB), SSIM, and IFC of different methods on image set “Kodak”

	2X			3X			4X		
	PSNR	**SSIM**	**IFC**	**PSNR**	**SSIM**	**IFC**	**PSNR**	**SSIM**	**IFC**
Bicubic	30.88	0.9787	5.46	28.46	0.8996	3.23	27.26	0.8438	2.13
NE+LLE	32.22	0.9937	6.81	29.21	0.9224	3.77	27.80	0.8711	2.45
ScSR	32.15	0.9923	6.65	29.07	0.9115	3.58	27.69	0.8707	2.32
ASDS	32.13	0.9930	6.79	29.11	0.9202	3.72	27.79	0.8715	2.51
ANR	32.28	0.9939	6.93	29.25	0.9233	3.82	27.84	0.8724	2.46
A+	32.75	0.9949	7.12	29.61	0.9278	3.90	28.13	0.8789	2.50
SRCNN	32.74	0.9934	7.09	29.39	0.9274	3.86	28.14	0.8777	2.42
VDSR	32.80	0.9950	7.19	29.59	0.9285	3.95	28.21	0.8790	2.53
GUN	**33.09**	**0.9959**	**7.27**	**29.73**	**0.9293**	**4.04**	**28.36**	**0.8811**	**2.76**

Reference

[1] R. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Trans. Acoustics, Speech Signal Process.* vol. 29, no. 6, pp. 1153-1160, Dec. 1981.

[2] T. M. Lehmann, C. Gonner, and K. Spitzer, “Survey: Interpolation methods in medical image processing,” *IEEE Trans. Med. Imag.*, vol. 18, no. 11, pp. 1049-1075, Nov. 1999.

[3] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong, “Softe edge smoothness prior for alpha channel super resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1-8.

[4] X. Li and M. T. Orchard, “New edge-directed interpolation,” *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521-1527, Oct. 2001.

[5] F. Zhou, W. Yang, and Q. Liao, “Interpolation-based image super-resolution using multisurface fitting,” *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3312-3318, Jul. 2012.

[6] D. Su and P. Willis, "Image interpolation by Pixel-Level Data-Dependent triangulation," *Comput. Graph. Forum*, vol. 23, no. 2, 2004.

[7] Q. Wang and R. K. Ward, “A new orientation-adaptive interpolation method,” *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 889-900, Apr. 2007.

[8] C. Zwart and D. Frakes, “Segment Adaptive Gradient Angle Interpolation,” *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 2960-2969, Aug. 2013.

[9] X. Liu, D. Zhao, R. Xiong, S. Ma, and W. Gao, “Image interpolation via regularized local linear regression,” *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3455-3469, Dec. 2011

[10] Q. Wang, R. Ward, and J. Zou, “Contrast enhancement for enlarged images based on edge sharpening,” in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, vol. 2, pp. 1-4.

[11] A. Giachett and N. Asuni, “Real-time artifact-free image upscaling,” *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2760-2768, Oct. 2011.

[12] X. Liu, D. Zhao, J. Zhou, W. Gao, and H. Sun, “Image Interpolation via Graph- Based Bayesian Label Propagation,” *IEEE Trans. Image Process.*, vol.23, no. 3, pp. 1084-1096, Mar. 2014.

[13] R.Fattal, “Image Upsampling via impose edge statistics,” *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007, Art. ID 95.

[14] J. Sun, Z. Xu, and H. Y. Shum, “Image super-resolution using gradient profile prior,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1-8.

[15] L. Wang, S. Xiang, G. Meng, et al, “Edge-Directed Single Image Super-Resolution via Adaptive Gradient Magnitude Self-Interpolation,” *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 23, no. 8, pp. 1289-1299, Aug. 2013.

[16] H. Xu, G. Zhai, and X. Yang, “Single image super-resolution with detail enhancement based on local fractal analysis of gradient,” *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 23, no. 10, pp. 1740-1754, Oct. 2013.

[17] L. Wang, H. Wu, and C. Pan, “Fast image upsampling via the displacement field,” *IEEE Trans. Image Process.*, vol.23, no. 12, pp. 5123-5135, Dec. 2014.

[18] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with non-local means and steering kernel regression,” *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4544-4556, Nov. 2012.

[19] Y. Zhao, R. Wang, W. Wang, and W. Gao, "High Resolution Local Structure-Constrained Image Upsampling," *IEEE Trans. Image Process.*, vol.24, no.11, pp.4394-4407, Nov. 2015

[20] A. Marquina, and S. J. Osher, “Image super-resolution by TV regularization and Bregman iteration,” *J. Sci. Comput.*, vol. 37, no. 3, pp. 367-382, Dec. 2008.

[21] X. Li, Y. Hu, X. Gao, D. Tao, and B. Ning, “A multi-frame image super-resolution method,” *Signal Process.*, vol. 90, no. 2, pp. 405-414, Feb. 2010.

[22] Q. Shan, Z. Li, J. Jia, et al, “Fast image/video upsampling,” *ACM Trans. Graph.*, vol. 27, pp. 32-39, 2008.

[23] T. Michaeli and M. Irani, “Nonparametric blind super-resolution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 945-952.

[24] N. Efrat, D. Glasner, A. Apartsin, et al., “Accurate blur models vs. image priors in single image super-resolution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2832- 2839.

[25] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, “Learning low-level vision,” *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 25-47, Jun. 2000.

[26] H. Chang, D. Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, vol. 1, pp. 275-282.

[27] M. Bevilacqua, A. Roumy, et al., “Low-complexity single image super-resolution based on nonnegative neighbor embedding,” in *Proc. British Machine Vis. Conf.*, 2012, pp.1-10

[28] M. Turkan, D. Thoreau, and P. Guillotel, “Optimized neighbor embeddings for single-image super-resolution,” in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 645-649.

[29] Y. Zhao, R. Wang, W. Wang, W. Gao, "Multilevel modified finite radon transform network for image upsampling," *IEEE Trans. Circuits and Syst. Video Technol.*, DOI: 10.1109/TCSVT.2015.2504731, 2015.

[30] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861-2873, Nov. 2010.

[31] J. Yang, Z. Wang, Z. Lin, X. Shu, and T. Huang, “Bilevel sparse coding for coupled feature spaces,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2360-2367

[32] W. Dong, D. Zhang, G. Shi, et al. “Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization,” *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1838-1857, Jul. 2011.

[33] L. He, H. Qi, and R. Zaretzki, “Beta Process Joint Dictionary Learning for Coupled Feature Spaces with Application to Single Image Super-Resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 345 - 352.

[34] T. Peleg and M. Elad, “A statistical prediction model based on sparse representations for single image super-resolution,” *IEEE Trans. Image Process.*, vol.23, no. 6, pp. 2569-2581, Jun. 2014.

[35] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Trans. Image Process.*, vol.22, no. 4, pp. 1620-1630, Apr. 2013.

[36] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” *Curv. Surfaces*, pp. 711- 730, 2010

[37] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, “A comprehensive survey to face hallucination,” *Int. J. Comput. Vis.*, vol. 106, no. 1, pp. 9-30, Jan. 2014.

[38] D. Glasner, S. Bagon, and M. Irani, “Super-resolution from a single image,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 349-356.

[39] G. Freedman, R. Fattal, “Image and video upscaling from local self-examples,” *ACM Trans. Graph.*, vol. 30, no. 2, pp. 12-23, 2011.

[40] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with multiscale similarity learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1648-1659, Oct. 2013.

[41] J. Yang, Z. Lin, S. Cohen, “Fast image super-resolution based on in-place example regression,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1059-1066.

[42] R. Timofte, V. D. Smet, and L. V. Gool, “Anchored neighborhood regression for fast example-based super-resolution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1920- 1927.

[43] R. Timofte, V. D. Smet, and L. V. Gool, " A+: Adjusted anchored neighborhood regression for fast super-resolution," *Asian Conf. Comput. Vis.*, Nov. 2014, pp. 1-15.

[44] C. Dong, C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Jul. 2016.

[45] J. Kim, J. Lee, and K, Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, vol. 1, pp. 1646-1654.

[46] J. Kim, J. Lee, K. Lee, “Deeply-Recursive Convolutional Network for Image Super-Resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, vol. 1, pp. 1637-1645.

[47] Y. Wang, L. Wang, H. Wang, P. Li, “End-to-End Image Super-Resolution via Deep and Shallow Convolutional Networks,” *arXiv preprint* arXiv:1607.07680, 2016.

[48] Z. Wang, D. Liu, Z. Wang, J. Yang, T. Huang, “Deeply improved sparse coding for image super-resolution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015.

[49] W. Shi, J. Caballer, F. Huszar, et al., “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, vol. 1, pp. 1874-1883.

[50] C. Dong, C. L. Chen, X. Tang, “Accelerating the Super-Resolution Convolutional Neural Network,” *arXiv preprint* arXiv:1608.00367, 2016.

[51] D. Liu, Z. Wang, N. Nasrabadi, et al., “Learning a Mixture of Deep Networks for Single Image Super-Resolution,” *arXiv preprint* arXiv:1701.00823, 2017.

[52] W. Yang, J. Feng, J. Yang, et al., “Deep Edge Guided Recurrent Residual Learning for Image Super-Resolution,” *arXiv preprint* arXiv:1604.08671, 2016.

[53] Z. Cui Z, H. Chang, S. Shan, et al., “Deep network cascade for image super-resolution,” in *Proc. Euro. Conf. Comput. Vis.*, 2014, pp.49-64.

[54] Y. Huang, W. Wang, L. Wang, “Bidirectional recurrent convolutional networks for multi-frame super-resolution,” in Proc. Conf. Neural Inform. Process. Syst., Dec. 2015, pp. 235-243.

[55] C. Ledig, L. Theis, F. Huszar, et al., “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” *arXiv preprint* arXiv:1609.04802, 2016.

[56] M. Sajjadi, B. Sch ölkopf, M. Hirsch, “EnhanceNet: Single Image Super-Resolution through Automated Texture Synthesis,” *arXiv preprint* arXiv:1612.07919, 2016.

[57] E. Denton, S. Chintala, A. Szlam, et al., “Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks,” in *Proc. Conf. Neural Inform. Process. Syst.*, Dec. 2015.

[58] A. Dosovitskiy, S. Tobias, T. Brox, “Learning to generate chairs with convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp.1538-1546.

[59] S. Schulter, C. Leistner, H. Bischof, “Fast and accurate image upscaling with super-resolution forests,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp.3791-3799.

[60] S. Gu, W. Zuo, Q. Xie, et al., “Convolutional Sparse Coding for Image Super-Resolution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp.1823-1831.

[61] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint* arXiv:1409.1556, 2014.

[62] A. Vedaldi and K. Lenc, "MatConvNet - Convolutional Neural Networks for MATLAB", in *Proc. of the ACM Int. Conf. on Multimedia*, 2015

[63] Z. Wang, A. Bovik, H. Sheikh, et al., “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, 2004.

[64] C. Yang, C. Ma, M. Yang, “Single-image super-resolution: a benchmark,” in *Proc. Euro. Conf. Comput. Vis.*, Sep. 2014, pp. 372-386.

[65] H. Sheikh, A. Bovik, G. De Veciana, “An information fidelity criterion for image quality assessment using natural scene statistics,” *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2117-2128, 2005.

[66] J. Salvador, E. Perezpellitero, “Naive Bayes Super-Resolution Forest,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 325-333.