A JOINT DEEP-NETWORK-BASED IMAGE RESTORATION ALGORITHM FOR MULTI-DEGRADATIONS

Anonymous ICME submission

ABSTRACT

In the procedures of image acquisition, compression, and transmission, captured images usually suffer from various degradations, such as low-resolution and compression distortion. Although there have been a lot of research done on image restoration, they usually aim to deal with a single degraded factor, ignoring the correlation of different degradations. To establish a restoration framework for multiple degradations, a joint deep-network-based image restoration algorithm is proposed in this paper. The proposed convolutional neural network is composed of two stages. Firstly, a de-blocking subnet is constructed, using a feedback recurrent neural network. Then, super-resolution is carried out by a 20-layer very deep network with skipping links. Cascading these two stages forms a novel deep network. Experimental results on the Set5 and Set14 benchmarks demonstrate that the proposed method can achieve better results, in terms of both the subjective and objective performances.

Index Terms— Image restoration**,** Joint Deep Network**,** Multi-degradations

1. INTRODUCTION

Due to the limitations of capturing devices and variations in lighting conditions, different kinds of degradations inevitably appear during the process of image capturing, coding, and transmission. Low-quality images not only have a negative effect on a human's visual experience, but also affect further automatic image analysis. In order to improve the image quality and the performance for image recognition, research on image restoration has become one of the most important issues in the field of image processing.

Over the years, many image super-resolution algorithms have been proposed. Learning-based approach is gradually becoming a hot research topic, which can be divided into shallow learning-based methods and deep learning-based methods.

For the shallow learning-based approaches, Freeman et al. [1] laid the foundation for the super-resolution (SR) restoration framework. After that, most of the research has been carried out based on this framework. However, this kind of algorithm usually requires a large training database. Each image patch from a low-resolution image searches in a large number of training samples for similar samples. Therefore, this approach is usually computationally intensive. A locally linear embedding method for manifold learning was introduced by Chang et al. in [2], which has the advantage of being more flexible for local detail reconstruction. However, this method has the problem of missing global constraints. In [3], a sparse-coding-based method was proposed, which can preserve the neighborhood relation and recover more detailed image information. However, in the sparse coding stage, it is computationally expensive, and is difficult to obtain an over-complete dictionary with strong representational ability. The above-mentioned method is greatly beneficial for selecting similar training samples from a large database to learn the models needed. However, the learning ability of shallow-learning-based methods is limited.

In the deep learning-based approaches, convolutional neural network (CNN) was firstly introduced to image super-resolution (SR) reconstruction by Dong et al. in [4], which is named as Super-Resolution using Convolutional Neural Network (SRCNN). Although SRCNN has a simple network structure，it can achieve an amazing restoration quality and a feasible runtime, compared to most previously described shallow learning-based methods. This shows that it is possible to combine traditional SR methods with deep-learning methods for solving the SR problem. After that, a deep convolutional network was introduced to image de-blocking by Dong et al. in [5]. In [5], a transfer learning strategy is used to fine-tune the SRCNN structure, and the reconstruction performance of compressed images is improved. This method shows that the deep network is also promising for the problem of image de-blocking. In order to enhance the network learning capability, an SR method based on Very Deep Networks (VDSR) was proposed by Kim et al. in [6], in which a cascade of low-order filters is used to develop the context information of larger image regions. This can further improve the reconstruction performance. In [7], Tezel et al. proposed a global-local up-sampling network (GLN), in which the network is pre-trained, by optimizing the reconstruction adversarial loss function used to adjust the network. In [8], a deep network cascade SR method (DNC) was proposed, which achieves the gradual enlargement of images by cascading several identical network structures.

These existing deep-learning methods mainly focus on the problem of one type of degradation. For example, ARCNN and VDSR are only effective for image de-blocking and SR, respectively. However, they cannot deal with the images that suffer from multiple degradations at the same time. [9] shows that image quality is mainly effected by resolution and compression. Image reconstruction of multi-degraded images, with the combination of low-resolution and JPEG compression distortion, is a great challenge. Therefore, a reconstruction method for images suffered from multi-degradations is proposed in this paper.

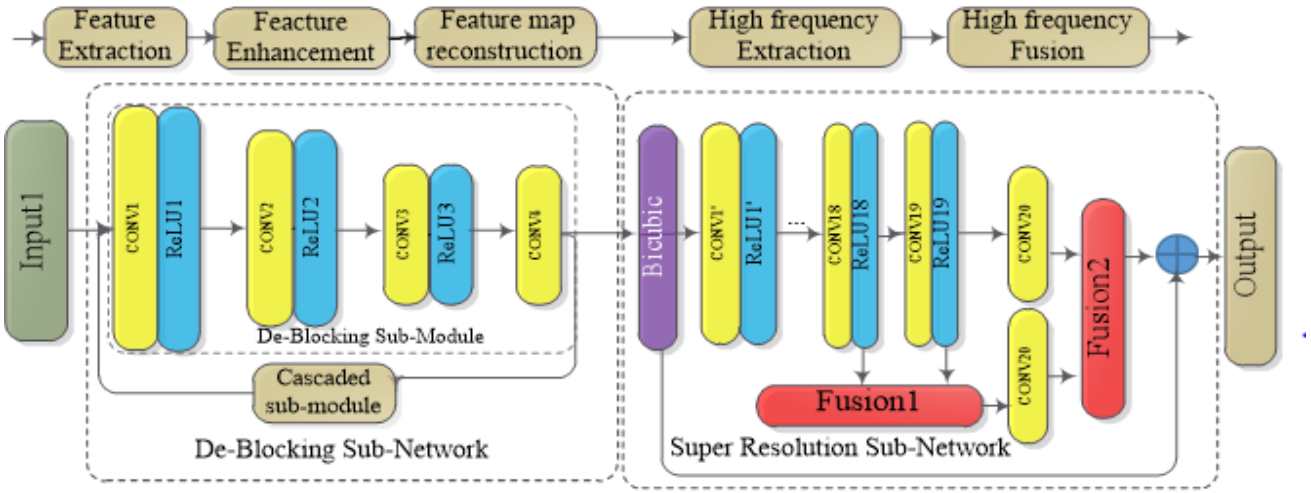


Fig. 1: The architecture of our proposed joint deep network. The network consists of 28 layers, which are a cascade of pairs of layers (convolutional and nonlinear). Left: A recurrent network with 8 convolutional layers to form the de-blocking sub-network. Right: The super-resolution sub-network first up-samples the input to an appropriate resolution by using the bicubic convolution layer. Then, a VGG19 is employed to extract the residual image features. Finally, the residual image features are enhanced by two fusion layers.

In this paper, we propose a joint deep-network-based image restoration method for low-resolution compressed images. Our framework can deal with both the compression artifacts and the low-resolution problem simultaneously. The main contributions of this paper are as follows: (1) A novel deep-network architecture with 28 convolution layers is designed for dealing with compressed artifacts and SR problems synergistically. (2) A recurrent structure is applied to the de-blocking subnet to remove the compressed artifacts. (3) Three skipping connections are added into a very deep network, with 20 convolution layers, to form the super-resolution subnet. The initial estimated image and three residual maps are fused to form the reconstructed HR image. The method can reduce the occurrence of blocking and ringing artifacts, as well as preserving more high frequency details. Experimental results show that the proposed algorithm can achieve better results, in terms of both subjective and objective performances, compared to state-of-the-art methods.

2. THE PROPOSED ALGORITHM

CNN is a deep neural network, suitable for continuous input signals, which has been successfully employed in different kinds of image restoration or enhancement problems. However, CNN is rarely used to deal with degraded images, in particular with multi-degradations, such as compressed, low-resolution images. It is necessary to establish a novel framework to restore compressed, low-resolution images.

The proposed deep-learning framework is shown in Figure 1. A joint deep network is designed, which consists of two subnets, namely a De-Blocking Sub-Network (DBSN) and a Super-Resolution Sub-Network (SRSN). The DBSN is used to remove the distortion appeared after decompression, and contains *K* identical, cascaded de-blocking (DB) modules. A DB module has three parts, feature extraction, feature enhancement, and feature map reconstruction, which work together for feature learning between noise feature maps and clean feature maps. The SRSN follows the DBSN, and contains two parts: high-frequency feature extraction and high-frequency feature fusion. High-frequency feature fusion is divided into two flows. One flow is for preserving the high-frequency information of the original network, while the other flow is used to fuse useful information between the inner layers.

2.1. De-Blocking Sub-Network

Existing single convolutional neural networks [4-7] are not good at dealing with multi-degradation images. In order to account for the compression artifact in multiple degradations, DB needs to be performed in reconstruction, which requires multi-level and stable feature learning. The network in [5] is employed as our sub-module, and the sub-modules are cascaded to form DBSN. Inspired by [8], a coarse-to-fine learning strategy is applied. The architecture of the DBSN is shown on the left of Figure 1. The DBSN consists of three DB modules connected in cascade, followed by a convolution layer (CONV4). Each DB module is formed by a convolution layer followed a RELU activation layer, which carries out the functions of feature extraction, feature enhancement, non-linear mapping, and feature reconstruction. The structure of the DBSN is summarized in Table 1.

Table 1. De-blocking sub-network architecture: First and second columns are the same DB sub-module. The first number after conv. indicates the kernel size, whereas the second number is the number of filters.

DB Sub-module	DB Sub-module
conv9-64	conv9-64
conv7-32	conv7-32
conv1-16	conv1-16
conv5-1	conv5-1
Concatenation	

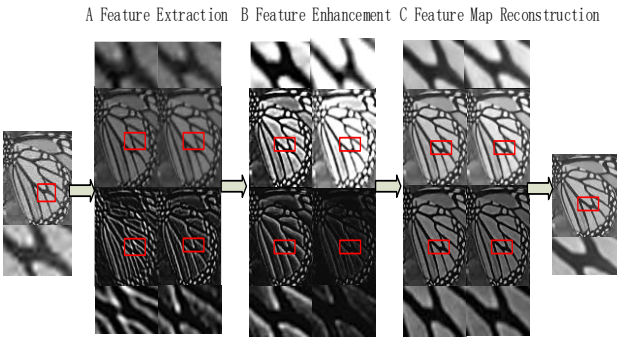


Fig. 2: Visualization of the feature maps at the different stages of the de-blocking sub-module (Q = 40).

Feature Extraction. An important starting point is the feature extraction step. Instead of traditional feature extraction, the image domain is associated with the feature domain by the convolution operations. In order to better accommodate the operations of the convolution filters, overlapping patches (i.e. 33×33 pixels) are extracted and adaptively adjusted between the different layers. Then, a new feature map is formed by a set of the patches, which is composed of a set of three-dimensional tensors. To further reduce the parameters of the network, the parameters of the convolution filters are adjusted by expert experience, which is equivalent to the traditional method Principal Component Analysis (PCA) for dimensionality reduction. In Figure 2, the input image is a degraded image, which is obtained by operating on the original image through two times of down-sampling and compressed via JPEG. Here, the quality parameter Q is set to 40. A variety of feature maps are obtained through different kinds of learned convolutional kernels.

The specific setting of this layer is listed as follows:

$$F_1(\mathrm{X}) = max(0, W_1 * \mathrm{X} + B_1), \qquad (1)$$

where W_1 and B_1 represent the filters and biases, respectively. W_1 is of size 1×9×9×64, which expresses 64 different convolutional filters. Each convolution filter has a kernel size 1×9×9. $F_1(\mathrm{X})$ is the feature map generated after the convolutional layer, which is composed of 64 feature maps. However, these feature maps contain some noises, which will affect subsequent feature learning. Thus, feature enhancement is essential for performance.

Feature Enhancement. Our work was inspired by ARCNN[5], where the feature map with noises is processed in the feature-map domain, and feature enhancement is achieved by adjusting the convolution filters. In fact, these operations are similar to combining feature maps to form another set of feature maps.

The specific setting of this layer is listed as follows:

$$F_2(\mathrm{X}) = max(0, W_2 * F_1(X) + B_2), \qquad (2)$$

where W_2 and B_2 represent the filters and biases, respectively. W_2 is of a size 64×7×7×32, it expresses 32 kinds of convolution filters, which each convolution has a kernel size 64×7×7. $F_2(\mathrm{X})$ is the feature map generated after the convolutional layer, which is composed of 32 feature maps.

In Figure 2, where the edge-enhanced convolution is used to process the extracted feature map, it can be observed that the edge features of the processed feature map are enhanced, and the noise can be reduced. Finally, a set of clean feature maps is obtained, but they are too bright or too dark.

Feature Map Reconstruction. The above two steps are achieved by using convolution operations. In order to prove these feature maps with non-linear characteristics and balance the brightness of the feature map, feature map reconstruction processing is necessary.

The specific setting of this step is listed as follows:

$$F_3(\mathrm{X}) = max(0, W_3 * F_2(\mathrm{X}) + B_3), \qquad (3)$$

where W_3 and B_3 represent the filters and biases, respectively. W_3 is of a size 32×1×1×16, and it expresses 16 kinds of convolution filters. The non-linear feature map is increased by a trivial spatial support 1×1, so each convolution has a kernel size 32×1×1. $F_3(\mathrm{X})$ is the feature map generated after the convolutional layer, which is composed of 16 feature maps.

$$F_4(\mathrm{X}) = W_4 * F_3(\mathrm{X}) + B_4, \qquad (4)$$

where W_4 and B_4 represent the filters and biases, respectively. W_4 is of size 16×5×5×1, which represents the convolution filters. Each convolutional filter has a kernel size of 16×5×5. $F_3(\mathrm{X})$ is the feature map generated after the convolutional layer, which is composed of one feature map.

In Figure 2, some overlapping patches are obtained in the feature-extraction phase. The overlapped regions are estimated via averaging. In order to better aggregate the patch-wise representations, the clean feature map is averaged by pre-defined filters.

Cascaded Network. In the above three steps, although a clean feature map is obtained by stable coarse-fine feature learning, it still suffers from noises and blocky artifacts. DBSN requires a deeper network, and in [8], the performance of the cascaded network architecture is superior to that of a single network. We consider both cases: directly deepening the number of layers will lose the stability in feature learning, and cascading the original de-blocking sub-module, which conducts stability fine feature learning. Our work was inspired by [8], where a cascaded network structure is employed, similar to the recurrent neural network. The subnet has K sub-modules, which are manually set according to the specific circumstances. K is set to 2 in our algorithm.

To train the network, the loss function Mean Squared Error (MSE) is given by:

$$\mathrm{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \|F(X_i; \theta) - Y_i\|^2, \qquad (5)$$

where X_i and Y_i are the i-th pair of low-resolution training data with or without compression, and $F(X_i; \theta)$ denotes the without compression image for predicted X_i using the de-blocking sub-network with parameter set θ, where $\theta = \{W_1; \ldots; W_4; B_1; \ldots; B_4\}$. To learn the SR network parameters θ, the loss function is minimized by using the back-propagation of the stochastic gradient descent method (SGD), with a fixed learning rate at 10^{-4} and the momentum at 0.9.

2.2. Super Resolution Sub-Network

The Super-resolution sub-network (SRSN) will produce a high-resolution target image. Although the deconvolution SR method [13] can achieve good results, the parameters of the deconvolution layers need to be specified for multi-scale magnification. To achieve image up-sampling, bicubic interpolation is used in the proposed method, which is also a convolution operation and can be formulated as a convolutional layer [4]. However, as the interpolated low-resolution image lacks high-frequency information, the combination of the two should be taken into account. In our proposed method, a special strategy of image information fusion is used, which includes three skipping connections into a very deep network, with 20 convolution layers to form the SRSN. The structure of the SRSN is summarized in Table 2.

Table2. Super-Resolution Sub-Network architecture. The first number before and after conv. indicate the name of the convolution layer and the kernel size, respectively, whereas the second number is the number of filters.

sub-module	layer name and parameter	
HE	1-19conv3-64	
HF	18conv3-64 19conv3-64	Fusion1
	20conv3-1	20conv3-1
	Fusion2	

Extraction of high-frequency features (HE). In the HE submodule, the quality of the reconstructed image is directly affected by the image's context details. To achieve better results, as inspired by Kim et al. [6], we use 19 identical convolution layers, which are formed by cascading a number of small-size filters. In these layers, the specific setting is listed as follows:

$$F_1(\mathrm{Y}) = W_1 * \mathrm{Y} + B_1, \quad (3)$$

$$F_{i+1}(\mathrm{Y}) = max(0, W_i * F_i(\mathrm{Y}) + B_i), \{i = 2, \dots 19\} \quad (4)$$

where W_i and B_i represent the 2-19th convolution filters and biases, respectively. The convolution filter W_1 is of a size 1×3×3×64, the other convolution filter W_i is of size 64×3×3×64, with both having the same kernel size of 64×3×3.

Feature Fusion (HF). In the HE submodule, the inter-layer information in the HE submodule needs to be integrated. Different fusion strategies will produce different results. Although [9] has joined information fusion, the result is slightly lower than that of [6], without information fusion. The reason for this is that low and high-level information is mixed in the reconstruction layer, and the result degrades slightly.

Based on these problems, our method uses a relatively simple fusion strategy, which mainly includes feature-map fusion and residual-image fusion. In Figure 1, the conv20 of the SRSN is the reconstruction layer, and the fusion layer was separately established before and after it, which achieves the role of feature enhancement and information compensation. Each layer in the HE submodule contains 64 different feature maps of the same size, where the features of the 18th and 19th layers are more abstract. The specific settings of the fusion layer (Fusion1) are as follows:

$$A(\mathrm{Y}) = G(F_{18}(\mathrm{Y}); F_{19}(\mathrm{Y})), \quad (5)$$

where F_{18}(Y) and F_{19}(Y) represent the 18th and 19th feature map, respectively, and G represents the feature-map fusion layer (Fusion1), which is a weighted convolution. In (6), by experiments, we set α at 0.4, Fusion2 uses the same setting.

$$A(\mathrm{Y}) = \alpha \times F_{18}(\mathrm{Y}) + (1 - \alpha) F_{19}(\mathrm{Y}). \quad (6)$$

The HF submodule is divided by two flows. One of the flows aims to protect the original high-frequency information, while the other flow aims to obtain more abstract information.

$$F_{19'}(\mathrm{Y}) = max(0, W_{20} * F_{19}(\mathrm{Y}) + B_{20}), \quad (7)$$

$$A_{19'}(\mathrm{Y}) = max(0, W_{20} * A(\mathrm{Y}) + B_{20}), \quad (8)$$

$$\mathrm{F} = G\left(F_{19'}(\mathrm{Y}); A_{19'}(\mathrm{Y})\right), \quad (9)$$

where $\mathrm{F}_{19'}$(Y) and $\mathrm{A}_{19'}$(Y) are the reconstruction residual image, with and without using feature-map fusion, and F denotes the result of residual-image fusion. The resulting high-resolution image R is generated as follows:

$$R = \mathrm{Y} + F. \quad (10)$$

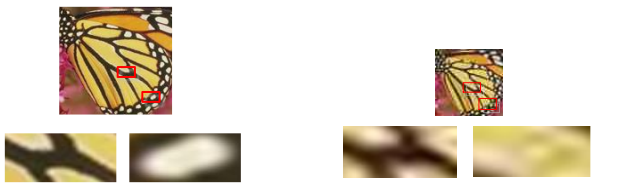
During training with a training dataset $\{Y^i, Z^i\}_{i=1}^N$, which represent the LR and HR training samples, respectively, we define the residual image $r = y–x$. Our goal is to minimize the loss function $\mathrm{L}(\theta) = \frac{1}{2}\|r - F(Y)\|^2$ using the back-propagation of the mini-batch gradient descent method (mini-batch) with a fixed learning rate schedule 10^{-1}, momentum of 0.9, and L2 penalty multiplied by 0.0001.

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

3.1. Experiment Settings

Datasets. Our models can be pre-trained using the training set in [10], which contains 91 images. Images in *Set5* [11] and *Set14* [12] are employed for testing, which contain 5 and 14 images, respectively. To train the deep network, a fine-tuning scheme is employed, based on the initial training network by the *ImageNet* [16] dataset. The zooming factor is set at 2 and 4. There are 91 images used for training and 14 images are used for testing.

Parameters. We first convert images from the RGB format into the YCbCr color format. HR training samples, LR training samples, and JPEG compressed samples are collected by randomly scaling, down-sampling and compression, respectively. In training Set 1, we generate patches of 41×41 pixels from the HR images in the training set, with a stride of 14 pixels. In training Set 2, we generate patches of 20×20 pixels from each LR images in the training set, with a stride of 10 pixels. In the de-blocking and super-resolution subnets, the proposed model is trained with the batch sizes of 128 and 64, respectively, and a fixed learning rate of 0.001 and 0.1 under different scaling factors.

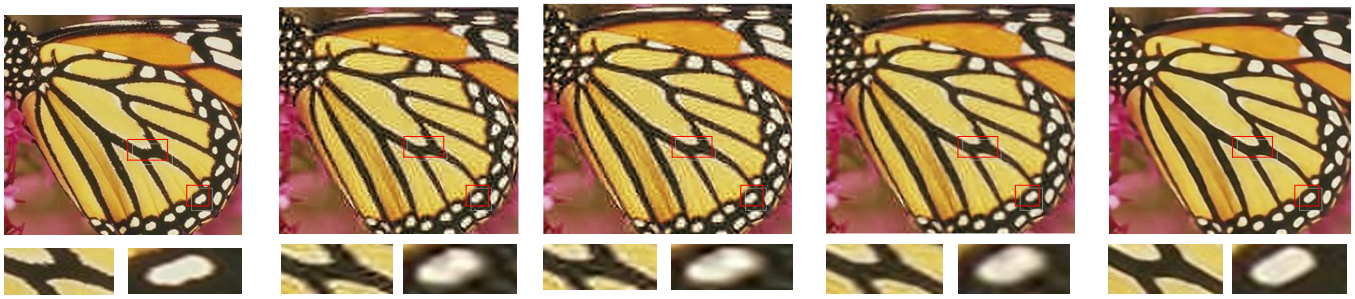


(a) (b)

Fig. 3: The input compressed low-resolution images. (a) Zooming factor (Z): 2, Compression quality(Q) is 40 (L2Q40), (b) Zooming factor: 4, Compression quality parameter: 20 (L4Q20).

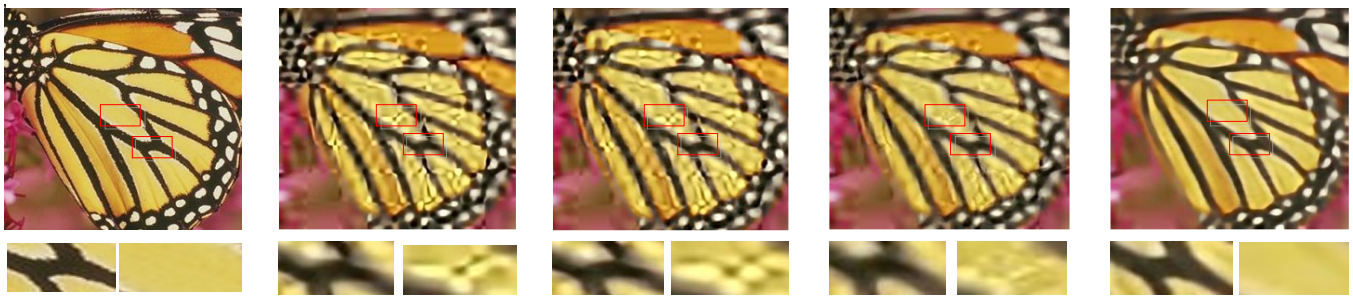
Table 3. Average PSNR results (dB) of different methods on the *Set5* and *Set14* datasets, with different Zooming factor (Z) and

Dataset	Z	Q	VDSR	FSRCNN	CSCN	Ours
Set5	2	40	29.52	29.48	30.39	**31.05**
Set5	4	20	24.45	24.54	25.10	**25.33**
Set14	2	40	26.97	27.40	27.31	**27.90**
Set14	4	20	23.07	23.41	23.36	**23.71**



(a) Original (b) VDSR[6] 24.35dB (c) FSRCNN[13] 24.45 dB (d) CSCN[14] 25.75dB (e) Ours 27.74 dB

Fig. 4: The results of different methods for the image 'Butterfly' (L2Q40).



(a) Original (b)VDSR[6] 18.97 dB (c) FSRCNN[13] 19.13 dB (d) CSCN[14] 19.75dB (e) Ours 20.71dB

Fig. 5: The results of different methods for the image 'Butterfly' (L4Q20).

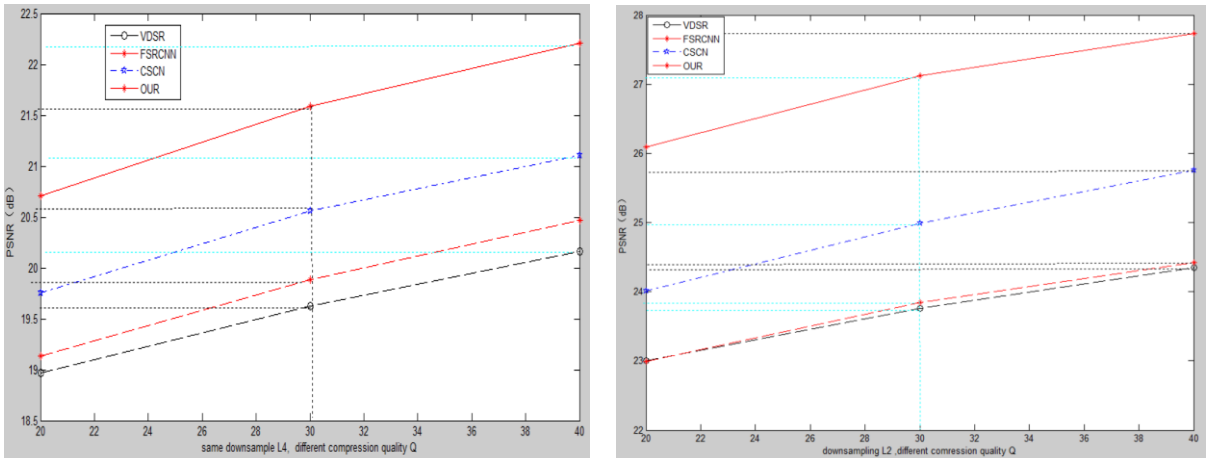
Evaluation Metric and Compared Methods. To evaluate the performance of the proposed method, we have compared it with three methods, including a very deep CNN (VDSR) [6], a fast SRCNN (FSRCNN) [13], and a SR with Sparse Prior (CSCN) [14]. We would like to thank the authors of [6], [13], [14] for providing the source code of their methods. In all the experiments, the original codes with default parameters are used. The configuration of the computer is Intel Core 3.6.GHz CPU and GTX1080 GPU in a MATLAB R2014a platform.

3.2. Experimental Results and Discussions

An input of compressed, low-resolution images with different compression ratios and different zooming factors are shown in Figure 3. The subjective performances of different deep-network-based image-restoration algorithms are shown in Figure 4 and Figure 5, respectively.

In Figure 4, from left to right shows the original images, the results of VDSR, FSRCNN [13], CSCN [14], and our method are illustrated. In the case of L2Q40, i.e. the compression quality is 40 and zooming factor is 2, compared with the other three methods, our proposed method significantly outperforms other methods separately in terms of subjective quality. Some methods exhibit ringing artifacts near the edges, but we can hardly perceive any ringing artifacts in the results produced by our method. From the zooming results in local regions, we can see that our method can preserve the original structures and remove almost all of the artifacts well.

In the case of L4Q20, due to the severe compression distortions and lower resolution, the three compared methods cannot handle the compressed artifacts well. From Figures 5(b), 5(c) and 5(d), we can see that the blocking artifacts have not been removed efficiently. From the zooming results, we can see that the artifacts not only destroy the textural structures of the original image, but also contaminate flat or smooth regions. In Figure 5 (e), the results of our method obviously outperform the compared results. The PSNR of each method is also listed under the respective images. With the different datasets, the best results are highlighted in Table 3, which shows that our approach is the best.



(a) The PSNR performances of 'Butterfly' (L4Q20) (b) The PSNR performances of 'Butterfly' (L2Q40)

Fig. 6: The PSNR performance of 'Butterfly' with different compression qualities

To show the objective results under different compression qualities and zooming factors, PSNRs for the image 'Butterfly' are plotted in Figure 6. We can see that our proposed method achieves the best performance, followed by FSRCNN, then CSCN, VDSR, etc. In other words, our method can deal with multiple degradations better than the compared methods.

4. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a joint deep network for compressed image super resolution. The architecture of the proposed method consists of two main components, namely a de-blocking sub-network and a super-resolution sub-network, which are cascaded and trained end to end, so that our method can tackle both the compression and low-resolution problems at the same time. The experimental results have shown that our method outperforms some state-of-the-art methods. In the future, the proposed model will be adapted to handle other low-level vision tasks.

5. REFERENCES

[1] Freeman W T, Jones T R, and Pastor E C, "Example-based super-resolution," Computer Graphics and Applications, *IEEE*, vol. 22, no 3, pp: 56-65, 2002

[2] Chang H, Yeung D Y, and Xiong Y, "Super-Resolution Through Neighbor Embedding," *IEEE* Computer Society Conference on Computer Vision and Pattern Recognition 2004, USA: Washington, DC, *IEEE* Press, 2004: I-275-I-282

[3] Yang J, Wright J, and Huang T. "Image Super-Resolution via Sparse Representation," *IEEE* Transactions on Image Processing, 2010, 99: 1-12

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

[5] Dong C, Y Deng, C C Loy and X Tang, "Compression Artifacts Reduction by a Deep Convolutional Network," *IEEE* International Conference on Computer Vision, 2015, vol. 71, no 2, pp:576-584

[6] Kim J, Lee J K, and Lee K M, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," *arXiv preprint arXiv*:1511 .04587, 2015.

[7] Tuzel O, Taguchi Y, and Hershey J R, "Global-Local Face Upsampling Network," *arXiv preprint arXiv*: 1603.07235, 2016.

[8] Z Cui, H Chang, S Shan, B Zhong and X Chen, "Deep Network Cascade for Image Super-resolution," In Computer Vision-*ECCV*. Springer International Publishing, 2014.

[9] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen., "Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment, "in Computer Vision *ECCV* 2014.

[10] Kim J, Lee J K, Lee K M, "Deeply-Recursive Convolutional Network for Image Super-Resolution," *arXiv preprint arXiv*:1511.04491, 2015.

[11] Yang, J., Wright, J., Huang, T., and Ma, Y, "Image super-resolution via sparse representation," *IEEE* Transactions on Image Processing 19(11) (Nov 2010) 2861–2873

[12] Bowden, R., Collomosse, J.P, and Mikolajczyk, K, eds.: "British Machine Vision Conference," *BMVA* (2012)

[13] Zeyde, R., Elad, M., and Protter, M, "On single image scale-up using sparse representations," In: Proceedings of the 7th International Conference on Curves and Surfaces, Berlin, Heidelberg, Springer-Verlag (2012)711–730

[14] Dong C, Loy C C, and Tang X, "Accelerating the super-resolution convolutional neural network [C]," European Conference on Computer Vision. Springer International Publishing, 2016: 391-407.

[15] Wang Z, Liu D, Yang J, "Deep Networks for Image Super-Resolution with Sparse Prior [C]," Proceedings of the *IEEE* International Conference on Computer Vision. 2015: 370-378.

[16] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., and Fei-Fei, L, "ImageNet: A large-scale hierarchical image database," In: *CVPR*. pp. 248–255 (2009)

[17] Zeyde, R., Elad, M., and Protter, M, "On single image scale-up using sparse representations," In: Proceedings of the 7th International Conference on Curves and Surfaces, Berlin, Heidelberg, Springer-Verlag (2012)7