This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

A Flexible Deep CNN Framework for Image Restoration

Zhi Jin, *Member, IEEE*, Muhammad Zafar Iqbal, *Student Member, IEEE*, Dmytro Bobkov, *Student Member, IEEE*, Wenbin Zou, *Member, IEEE*, Xia Li, *Member, IEEE*, Eckehard Steinbach, *Fellow, IEEE*

***Abstract*—Image restoration is a long-standing problem in image processing and low-level computer vision. Recently, discriminative convolutional neural network (CNN)-based approaches have attracted considerable attention due to their superior performance. However, most of these frameworks are designed for one specific image restoration task; hence, they seldom show high performance on other image restoration tasks. To address this issue, we propose a flexible deep CNN framework that exploits the frequency characteristics of different types of artifacts. Hence, the same approach can be employed for a variety of image restoration tasks by adjusting the architecture. For reducing the artifacts with similar frequency characteristics, a quality enhancement network that adopts residual and recursive learning is proposed. Residual learning is utilized to speed up the training process and boost the performance; recursive learning is adopted to significantly reduce the number of training parameters as well as boost the performance. Moreover, lateral connections transmit the extracted features between different frequency streams via multiple paths. One aggregation network combines the outputs of these streams to further enhance the restored images. We demonstrate the capabilities of the proposed framework with three representative applications: image compression artifacts reduction (CAR), image denoising, and single image super-resolution (SISR). Extensive experiments confirm that the proposed framework outperforms the state-of-the-art approaches on benchmark datasets for these applications.**

***Index Terms*—Image restoration, Flexible CNN framework, Image decomposition, Recursive learning, Residual learning.**

I. Introduction

Image restoration (IR), as one of the most fundamental tasks in image processing and low-level computer vision, aims to reconstruct the latent high-quality (HQ) image from its distorted observation [1]. Degradation can arise from coding artifacts, resolution limitations, transmission noise, object motion, and camera movement or a combination of them. Accordingly, IR includes image compression artifacts reduction (CAR), image denoising, single image super-resolution (SISR), deblurring, dehazing, etc.

Due to the capability of providing state-of-the-art performance for high-level computer vision problems, deep learning (DL)-based methods have become a more recent trend to solve the IR problem [3]. Moreover, by focusing on learning a nonlinear mapping between the distorted images and their corresponding HQ images, regression-type neural networks have demonstrated impressive results on inverse problems with exact models [4].

The first DL-based solutions to SISR and image CAR tasks were the SRCNN [5] and ARCNN [6], respectively. They have demonstrated the effectiveness of CNNs by solely adopting shallow networks. However, due to the limited representation capacity of shallow networks, they still suffer from oversmoothing the reconstructed images. As a deep network, although DnCNN [7] is designed for image denoising, it shows promising performance on image CAR and SR as well. The end-to-end deep networks RED30 [8], ARN [9], MemNet [10] and MWCNN [11] target solving the IR problem; however, all of them treat all types of artifacts equally. Therefore, the different characteristics of various artifacts are not taken into account. Unfortunately, without specific consideration of various artifacts, one can observe the following issue. Specifically, the reduction of one type of artifact can lead an unintentional increase in other types of artifacts [3, 12]. In addition, while seeking to fulfill "the deeper the better" premise, most of the deep networks suffer from high computational cost due to an enormous number of training parameters. For example, RED30 and ARN have 4,131k and 1,145k training parameters, respectively, while the number of training parameters of our network for image SR is only 594k.

Based on the analysis of the three main degradations: coding artifacts (or compression artifacts), resolution limitations, and transmission noise, we found that all introduced artifacts can be classified into either a high-frequency (HF) or a low-frequency (LF) group. For the example of image CAR, we take JPEG as the compression algorithm. By adopting the block-based discrete cosine transform (BDCT) together with coarse quantization of the low spectral coefficients, JPEG compression causes blocking artifacts at the 8×8 block borders and ringing artifacts in the smooth portions of the images [13]. Moreover, the truncation of high-frequency DCT coefficients introduces blurring artifacts. While blocking and

This work was supported by the National Natural Science Foundation of China (No. 61701313, No. 61771321, No. 61871273, No. 61872429), China Postdoctoral Science Foundation Grants (No. 2017M622778), Guangdong Key Research Platform of Universities (No. 2018WCXTD015), the Science and Technology Program of Shenzhen (No. JCYJ20170818091621856) and the Interdisciplinary Innovation Team of Shenzhen University.

Corresponding author: Wenbin Zou.

Zhi Jin, Wenbin Zou and Xia Li are with the Guangdong Key Laboratory of Intelligent Information Processing, College of Electronics and Information Engineering, Shenzhen University, P.R. China. Wenbin Zou is also with the Shenzhen Key Laboratory of Advanced Machine Learning and Applications. E-mail: jinzhi_126@163.com, wzou@szu.edu.cn, lixia@szu.edu.cn

Muhammad Zafar Iqbal, Dmytro Bobkov and Eckehard Steinbach are with the Chair of Media Technology, Technical University of Munich, Munich, Germany. E-mail: mzafar.iqbal@tum.de, dmytro.bobkov@tum.de, eckehard.steinbach@tum.de

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

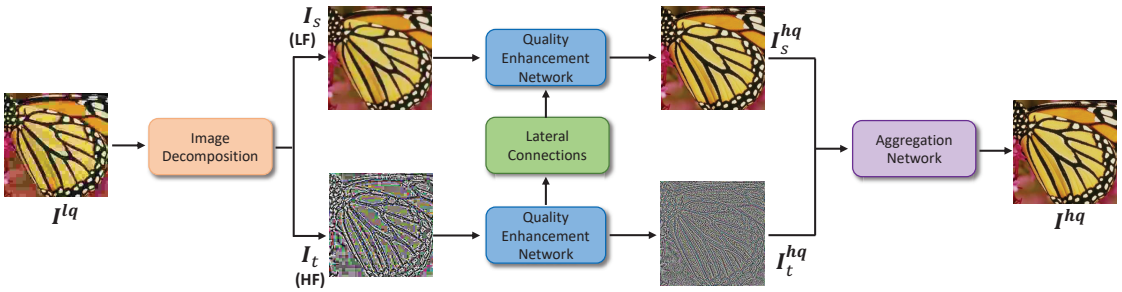


Figure 1. Framework of the proposed flexible deep network architecture, where each component of the framework can be removed if the distorted image only suffers from HF/LF artifacts. Here, we use the compression artifact reduction process as an example.

ringing artifacts belong to the HF artifacts group, blurring artifacts belong to the LF artifacts group. These two artifact groups can be successfully separated by decomposing the compressed images into an HF and LF component, as well. Therefore, to reduce the artifacts, the HF and LF components of the compressed image should be manipulated separately. The target of HF component processing is to reduce the blocking and ringing artifacts, and the target of LF component processing is to reduce the blurring artifacts. For image denoising, the noise is involved in the HF component of the noisy images. Moreover, it is important to note that some blurring artifacts are introduced into the LF component after noise extraction. Thus, the target of LF component processing is also to reduce the blurring artifacts. In terms of the image SR, super-resolved images suffer from blurring artifacts that are caused by inaccurate estimations of the missing pixels during the interpolation process [14]. Hence, in this case, the target for image SR can be regarded as reducing the blurring artifacts introduced by interpolation. We believe that better IR performance can be obtained on these tasks by specifically and separately processing the HF and LF artifacts.

Motivated by the above observation, in this work, we propose a flexible deep CNN framework for image restoration, especially image CAR, denoising and SR. The whole framework consists of four modules: image decomposition, a quality enhancement network on each decomposed part, lateral connections and an aggregation network. Each of them has an explicit function and can be removed from the whole framework depending on the specific task. Image decomposition is used for decomposing the distorted image into a texture layer (HF component) and a structure layer (LF component) with the goal of separating HF and LF artifacts. The quality enhancement network aims to enhance the quality of the corresponding image component by reducing the HF/LF artifacts. The lateral connections are developed for progressively transmitting the HF information from the texture layer to the structure layer in order to boost the enhancement process on the structure stream. The enhanced texture and structure layers are combined and fed into an aggregation network to generate the final enhanced image. Hence, the proposed work is different from [15–17], where only the HF component is enhanced by learning-based approaches and the final images are obtained by directly adding the enhanced HF component back to the corresponding LF component. Additionally, multipath residual learning and recursive learning are adopted in the quality enhancement network to speed up network training by easing this process [7, 18–21] and reduce the number of training parameters. The residual units in the same quality enhancement network share weights. Hence, the number of training parameters of the quality enhancement network is fixed and equivalent to that of a 4-layer CNN.

The main contributions of the proposed work are summarized as follows:

(1) **We propose a flexible deep CNN framework for IR that can be easily adapted to a specific IR problem by simply removing one or more certain modules.** In this way, each IR problem can be handled by a particular framework.

(2) **We combine global, local and intermediate residual learning with recursive learning to form multipath recursive residual learning in the quality enhancement network.** This combination is novel and allows helping not only the gradient flow but also the transmission of low-level features. In particular, benefiting from the shared weights between each residual unit, the network with fewer parameters becomes more compact. Furthermore, the number of training parameters will not increase with the network depth.

(3) **We set up lateral connections between the texture and structure streams, which allows transmitting the features between the two streams.** The lateral connections transmit the extracted features at different levels from one stream to the other via multiple paths. Due to weight sharing, the number of parameters of all these paths is equivalent to that of a 1-layer CNN.

The remainder of this paper is organized as follows. In Sec. II, we describe the related work about image restoration from three perspectives. Sec. III introduces the details of each component in our flexible framework. The experimental results are presented in Sec. IV, showing significant improvements for the proposed idea in image CAR, Gaussian image denoising, and SISR. Ablation analysis is provided as well. Sec. V concludes the paper and describes our future work.

II. Related Work

For several decades, image restoration (IR) has remained an active research topic, and plenty of approaches have been proposed in the literature. These approaches can be classified

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

into image prior-based [22], example-based [23], dictionary-based [24, 25], transformed domain-based [26, 27] and the currently most popular DL-based approaches [6, 28]. Our work is inspired by the state-of-the-art works on IR, especially those based on DL. Therefore, we focus on discussing the related works on DL-based JPEG image CAR, image denoising and SISR.

A. JPEG Compression Artifacts Reduction

Dong et al. [6] were the first to introduce a convolutional neural network (i.e., ARCNN) to learn an end-to-end mapping between compressed and restored images. Although ARCNN has demonstrated the effectiveness of deep learning on JPEG CAR by merely using a 4-layer network, due to the limited capacity of a shallow network, the reconstructed images still lack sharp edges and have overly smoothed texture regions. Nevertheless, they encountered difficulties in obtaining better performance with a deeper network by simply stacking the convolutional layers for such a low-level vision task. However, the training difficulties have been mitigated by the newly proposed network designs, e.g., local residual learning (LRL) [20], where the residual learning is performed locally within each residual unit, and global residual learning (GRL) [28], where the residual learning is performed directly between the input and the output. Although the network in [11] achieves superior performance compared to ARCNN, the number of involved training parameters is up to 16,140k. By applying DCT-domain prior knowledge of JPEG compression to the pixel-domain, *Guo et al.* [30] built a dual-domain 20-layer network with 1,114k training parameters. However, while the DCT-domain prior knowledge improves the performance of JPEG CAR, it limits its application to other compression algorithms, e.g., JPEG2000. Moreover, since JPEG compression causes both HF and LF artifacts in the distorted images, to the best of our knowledge, the proposed work is the first to adopt DL networks to separately reduce the artifacts based on their frequency characteristics.

B. Image Denoising

Chen et al. [31] proposed a trainable nonlinear reaction diffusion (TNRD) model that can be expressed as a feedforward deep network by unfolding a fixed number of gradient descent inference steps. *Burger et al.* [32] successfully applied a plain multilayer perceptron (MLP) to remove the noise, which can achieve promising performance and is able to compete with the previous state-of-the-art denoising method BM3D [33]. *Xie et al.* [34] combined sparse coding and deep networks to handle Gaussian noise removal. *Zhang et al.* [7] proposed a deep residual neural network called DnCNN. Different from the existing discriminative denoising models that usually train a specific model for additive white Gaussian noise (AWGN) at a certain noise level, DnCNN is able to handle Gaussian denoising with an unknown noise level. However, DnCNN heavily relies on a massive number of training images, which are added with various noise levels in the range $[0, 55]$. Although TNRD and DnCNN are designed for image denoising, they show good performance on image CAR and SISR, as well.

C. Image Super-resolution

DL-based SR solutions have gained increasing research interest in recent years. *Dong et al.* [5] utilized a 3-layer fully convolutional network (SRCNN) to learn the nonlinear mapping between HR and LR patches. *Kim et al.* [28] proposed a 20-layer convolutional network (VDSR) with GRL that significantly improves the reconstruction performance. *Shi et al.* [35] proposed a contextualized multitask convolutional network to super-resolve images while well preserving the structural details. However, the main drawback of these deep networks is that the number of learned parameters linearly increases with the network depth. To address these issues, on the one hand, *Kim et al.* [36] introduced a recursive layer into the network DRCN for image SR, where the same weights are applied to feature maps recursively. Therefore, the training parameters do not increase when more recursions are performed in the recursive layer. On the other hand, *Lai et al.* [37] proposed a Laplacian pyramid super-resolution network (LapSRN) to progressively reconstruct the subband residuals of high-resolution images. Since the network takes low-resolution images as the input, the computational complexity is significantly reduced. However, due to the pyramid structure, LapSRN is hard to extend to solve other IR tasks. A similar problem is also encountered by EDSR [38], RDB [39], RCAN [40] and IDN [41], where an upsampling network is adopted at the end of the proposed framework to obtain the final recovered high-resolution images.

III. Proposed Framework

In this section, for simplicity we consider JPEG image compression as an example, which yields both HF and LF artifacts. We do so to explain the function of each module of the proposed framework. Then, the modified frameworks for image denoising and SR will be introduced. Without loss of generality, we assume that the input image is a single-channel grayscale image. The proposed approach can be easily extended to common RGB images by repeating the proposed process for each color channel.

A. Problem Statement

Denote $\hat{\mathbf{x}}$ as the latent image (the ground truth image) and $\mathbf{y}$ as the degraded observation of it. A typical image degradation model can be written as:

$$\mathbf{y} = \boldsymbol{H}\hat{\mathbf{x}} + \mathbf{N} \tag{1}$$

where $\boldsymbol{H}$ denotes the degradation operator and $\mathbf{N}$ denotes the additional noise. Different IR problems can be defined based on the form of $\boldsymbol{H}$. For example, in image denoising $\boldsymbol{H}$ is an identity matrix. In image deblurring, $\boldsymbol{H}$ is a blurring operator. In image SR, $\boldsymbol{H}$ is a composition operator of blurring and downsampling. Therefore, the solution to IR can be expressed as obtaining an estimation $\mathbf{x}$ by minimizing the objective function as follows:

$$\mathbf{x} = \arg\min_{\hat{\mathbf{x}}} \{\|\mathbf{y} - \boldsymbol{H}\hat{\mathbf{x}}\|_2^2\} \tag{2}$$

The linear system in Eq. (1) is generally ill-posed, i.e., we cannot obtain $\mathbf{x}$ by directly solving Eq. (2). To address this

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

problem, traditional image restoration methods usually employ regularization techniques by adding some constraints to derive the estimation in Eq. (2) as follows:

$$\mathbf{x} = \arg\min_{\hat{\mathbf{x}}} \|\mathbf{y} - \boldsymbol{H}\hat{\mathbf{x}}\|_2^2 + \lambda R(\hat{\mathbf{x}}) \tag{3}$$

where λ is a Lagrange multiplier directly controlling the significance of the regularity term $R(\hat{\mathbf{x}})$. Therefore, we can estimate the unknown latent image $\hat{\mathbf{x}}$ from its observation $\mathbf{y}$.

Referring to Fig. 1, the proposed framework starts with the decomposition of the degraded images, in order to ensure that the LF and HF artifacts are contained within the corresponding components of images. Therefore, the input image is formulated as the superposition of a structure component (LF component) and a texture component (HF component), i.e.,

$$\mathbf{I}^{lq} = \mathbf{I}_s^{lq} + \mathbf{I}_t^{lq} \tag{4}$$

where $\mathbf{I}^{lq}$ is the low-quality image, $\mathbf{I}_s^{lq}$ is the structure component corresponding to the smooth areas of the image, and $\mathbf{I}_t^{lq}$ is the texture component corresponding to the fine details. The structure and texture components are processed separately by the proposed quality enhancement network with different targets. Since the decomposition is linear, Eq. (3) is equivalent to

$$\mathbf{I}_s^{hq} = \arg\min_{\hat{\mathbf{I}}_s^{hq}} \{\|\mathbf{I}_s^{lq} - \boldsymbol{H}_s\hat{\mathbf{I}}_s^{hq}\|_2^2 + \lambda_s R(\hat{\mathbf{I}}_s^{hq})\} \tag{5a}$$

$$\mathbf{I}_t^{hq} = \arg\min_{\hat{\mathbf{I}}_t^{hq}} \{\|\mathbf{I}_t^{lq} - \boldsymbol{H}_t\hat{\mathbf{I}}_t^{hq}\|_2^2 + \lambda_t R(\hat{\mathbf{I}}_t^{hq})\} \tag{5b}$$

Similarly, we can obtain the estimated high-quality image $\mathbf{I}_s^{hq}$ and $\mathbf{I}_t^{hq}$ for the unknown latent image $\hat{\mathbf{I}}_s^{hq}$ and $\hat{\mathbf{I}}_t^{hq}$ from their observation $\mathbf{I}_s^{lq}$ and $\mathbf{I}_t^{lq}$, respectively. By learning mapping parameters Θ through an optimization of the loss function $\mathcal{L}$ on training data, the objective function for the proposed quality enhancement network can be expressed as:

$$\min \mathcal{L}(\Theta_s) = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathbf{I}}_{s_i}^{hq} - \mathcal{H}(\mathbf{I}_{s_i}^{lq}, \Theta_s)\|_2^2 \tag{6a}$$

$$\min \mathcal{L}(\Theta_t) = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathbf{I}}_{t_i}^{hq} - \mathcal{H}(\mathbf{I}_{t_i}^{lq}, \Theta_t)\|_2^2 \tag{6b}$$

where $\mathcal{H}$ is the mapping function of the quality enhancement network. Therefore, the original regularization-based solutions ((5a) and (5b)) can be solved, respectively, by finding the optimal solutions on the structure and texture components based on the proposed DL network ((6a) and (6b)). In the following, the details of each part are discussed.

B. Image Decomposition

The problem of extracting the HF artifact-free structure image from the degraded image is also ill-posed. To obtain an accurate estimation, the decomposing step is posed as a total-variation (TV)-based structure extraction problem described in [42]. Therefore, the optimal structure image is obtained first. Then, the texture image is obtained as the difference between the compressed image and the structure image (see Fig. 2 for an example).

Let $\mathbf{I}^{lq}(x, y)$ be denoted as each pixel on the observed compressed image and $\mathbf{I}_s^{lq}$ be denoted as the optimal structure image, which is free from high-frequency compression artifacts. Our constrained minimization problem can be expressed as:

$$\mathbf{I}_s^{lq} = \arg\min_{\tilde{\mathbf{I}}_s^{lq}} \mathrm{J}(\tilde{\mathbf{I}}_s^{lq}) \tag{7}$$

with

$$\mathrm{J}(\tilde{\mathbf{I}}_s^{lq}) = \frac{1}{2}\|\mathbf{I}^{lq} - \boldsymbol{H}\tilde{\mathbf{I}}_s^{lq}\|_2^2 + \lambda_s \mathrm{TV}(\tilde{\mathbf{I}}_s^{lq}) \tag{8}$$

where $\tilde{\mathbf{I}}_s^{lq}$ represents a possible structure image and is blurred by a blurring operator $\boldsymbol{H}$ [43]. Since λ_s controls the weight of the regularity term, it needs to be adjusted according to the compression factor. Higher compression requires a larger λ_s, which guarantees the structure component to be free from HF artifacts. Eq. (8) can be solved by the half-quadratic splitting algorithm [44] based on the idea of introducing auxiliary variables to expand the original terms and update them iteratively.

C. Quality Enhancement Network

After image decomposition, the HF and LF artifacts are successfully separated and can be reduced along with the corresponding image components. Hence, we develop a quality enhancement network to fulfill this task. Inspired by [20] and [28], we adopt LRL (Fig. 3(a)) and GRL (Fig. 3(b)) in the proposed network. In addition, we propose a multipath intermediate residual learning (IRL) between the GRL and LRL to further help the gradient flow and the transmission of low-level features. Furthermore, to address the high computational and storage cost caused by the enormous number of training parameters of deep networks, recursive learning [36] is adopted to tackle this problem. Therefore, the quality enhancement network is in a multipath recursive residual learning structure (Fig. 3(c)). The basic unit of the proposed network is called the residual unit (RU), and several units that are stacked together form one recursive block (RB) (shown in purple in Fig. 3(c)). In the following, the architecture and training settings are given in detail.

1) Residual Unit: The involved residual unit has the same architecture as ResNet but relies on an opposite activation order, which moves the activation layers (BN and ReLU) before the convolutional layer. This modification has been validated to be more efficient in network training and can achieve better performance than the original activation architecture [45]. Due to the maintenance of the original chain structure of the residual unit, all the advantages in ResNet e.g., fast convergence, are well kept in the proposed network architecture. The formulation of one residual unit can be expressed as:

$$\begin{aligned}\hat{\mathbf{x}}^u &= \mathcal{F}(\mathbf{x}^u, W^u) + \mathbf{x}^u \\ &= f_2^u\left(\sigma(f_1^u(\sigma(\mathbf{x}^u), W_1^u)), W_2^u\right) + \mathbf{x}^u\end{aligned} \tag{9}$$

where $\mathbf{x}^u$ and $\hat{\mathbf{x}}^u$ are the input and output of the u-th residual unit, respectively, $\mathcal{F}$ denotes the residual mapping function for one residual unit, f_i^u is the mapping function of the i-th convolutional layer in the u-th unit, W_i^u represents a set

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

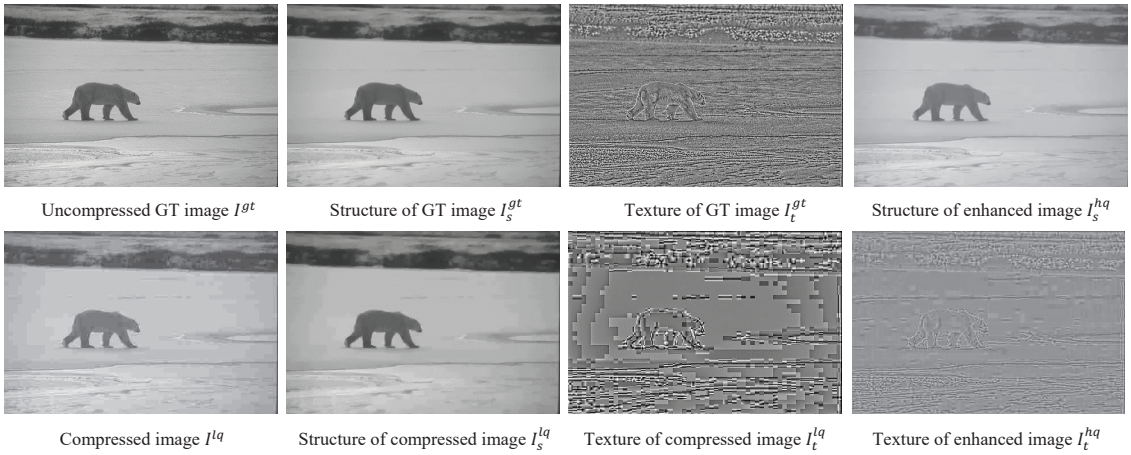


Figure 2. Image decomposition example for an uncompressed and compressed image (QF10) pair (λ_{gt} = 0.02 and λ_s = 0.04). We can notice that most of the blocking and ringing artifacts exist in the texture component of the compressed image, while most of the blurring artifacts exist in the structure component. The last column shows the structure and texture components of the enhanced image obtained by the proposed quality enhancement network.

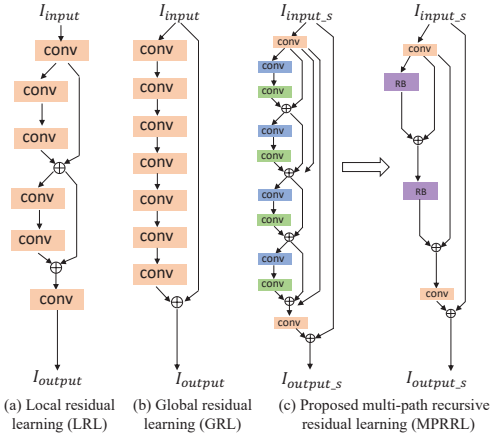


Figure 3. Simplified (a) LRL [20] network architecture shown with two residual units; (b) GRL [28] network architecture; (c) the proposed multipath recursive residual learning (MPRRL) network shown with two recursive blocks, each of which includes two residual units sharing the same weights (represented in light green and blue).

of weights (the biases are omitted for simplicity) in the i-th convolutional layer, and function σ denotes ReLU activation.

2) Recursive Block: We denote the function of each recursive block as $\mathcal{B}$, the corresponding input and output for the b-th recursive block are $\mathbf{x}^b$ and $\hat{\mathbf{x}}^b$, respectively, and W_i^b represents the weights of the i-th residual unit in the b-th recursive block. Therefore, the example shown in Fig. 3(c), where one recursive block consists of two residual units, can be expressed as:

$$\hat{\mathbf{x}}^b = \mathcal{B}(\mathbf{x}^b) = \mathcal{F}(\mathcal{F}(\mathbf{x}^b, W_1^b) + \mathbf{x}^b, W_2^b) + \mathcal{F}(\mathbf{x}^b, W_1^b) + 2\mathbf{x}^b \quad (10)$$

From Eq. (10), we observe that the architecture of the proposed recursive block passes the output of each intermediate residual unit to the end of the recursive block, which can well indicate the low-level features have been passed to the deep convolutional layers.

In the given example in Fig. 3(c), the proposed network with two recursive blocks in a multipath mode can be expressed as:

$$\hat{\mathbf{x}} = \mathcal{H}(\mathbf{x}) = f_{rec}(\mathcal{B}(\mathcal{B}(f(\mathbf{x})) + f(\mathbf{x})) + f(\mathbf{x})) + \mathbf{x} \quad (11)$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the input and output of the network, respectively, f and f_{rec} are the mapping functions of the first and last convolutional layers, and $\mathcal{H}$ represents the mapping function of the proposed quality enhancement network.

Given a training set $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where N is the number of training patches and $\mathbf{y}_i$ is the ground truth patch of the low quality patch $\mathbf{x}_i$, the loss function of the proposed network is

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \|\mathcal{H}(\mathbf{x}_i, \Theta) - (\mathbf{y}_i - \mathbf{x}_i)\|_2^2 \quad (12)$$

where Θ denotes the network parameter set. The training ground truth patches for the structure and texture streams are obtained from the uncompressed images and the corresponding texture layers, respectively.

3) Flexible Network Structure: The proposed quality enhancement network has a flexible architecture as well, which benefits from the three kinds of residual learning and the designs of the RU and RB. Given one specific depth, the number of residual units denoted as U, and the number of recursive blocks denoted as B can be freely adjusted. Specifically, the depth of the proposed network is calculated as:

$$d = 2 + 2 \times U \times B \quad (13)$$

If d = 20, the network has the following three versions: 1B9U (9 residual units in only one recursive block), 3B3U (3 residual units in each recursive block, and 3 recursive blocks in total) and 9B1U (only one residual unit in each recursive block, and 9 recursive blocks in total), referring to Fig. 4. The 1B9U version could be regarded as ResNet combined with GRL. Only the 3B3U version contains three kinds of residual learning. Therefore, the 20-layer quality enhancement network in the 3B3U architecture is applied separately to the structure and texture streams in our proposed flexible framework. It is worth noting that the deeper the network is, the more flexible the network architecture will be.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

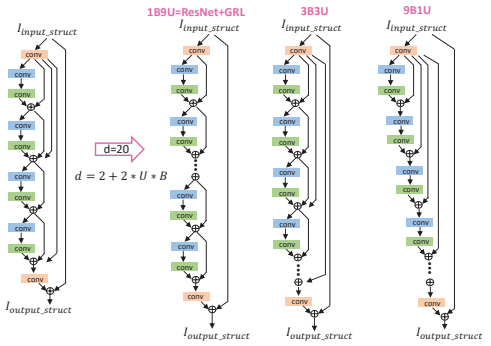


Figure 4. An example to show the flexible architecture of the proposed quality enhancement network with network depth 20.

D. Lateral Connections

Along with multiple nonlinear mappings, the structure and texture images are enhanced progressively, and the corresponding outputs are combined to generate preliminary recovered high-quality images. Instead of combining the enhanced images at the end, lateral connections are applied to transmit the improved HF information at each level from the texture stream to the counterpart of the structure stream, which can boost up the performance of HF information recovery in the structure stream.

A detailed example of lateral connections is illustrated in Fig. 5, where each path of the lateral connections is formed by a single convolutional layer with ReLU. The connections transform the extracted feature maps $\hat{x}_t^u$ from the texture stream to the lateral feature maps $\hat{x}^l$ after each residual unit. Then, these lateral feature maps are combined in a pixelwise manner with the corresponding feature maps on the structure stream $\hat{x}_s^u$. They can thus continue the following nonlinear mappings on the structure stream. To keep the whole network compact, we apply the weight sharing strategy. As a result, the total number of training parameters of all the lateral connections is equivalent to that of a single layer CNN.

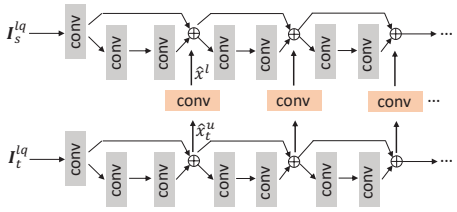
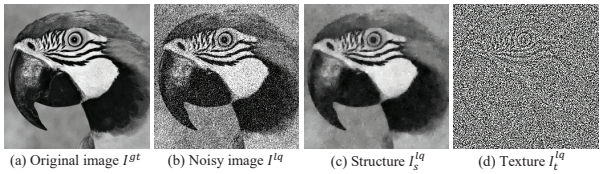


Figure 5. Structure of lateral connections.

E. Aggregation Network

The structure stream and the texture stream are trained together with lateral connections, but separately output an enhanced version of the original corresponding image component, denoted as $\mathbf{I}_s^{hq}$ and $\mathbf{I}_t^{hq}$, respectively. Then, the corresponding outputs are added in a pixelwise manner to form an enhanced input image, i.e., $\mathbf{I}_{agg}^{hq} = \mathbf{I}_s^{hq} + \mathbf{I}_t^{hq}$. Finally, $\mathbf{I}_{agg}^{hq}$ is fed into a nonlinear aggregation network, whose architecture is the same as the quality enhancement network to further improve the restored image. The patches obtained from the uncompressed images are used to supervise the training process of the aggregation network. At this stage, a dual-stream 40-layer image restoration framework is formed for the image CAR and named "Pro-CAR" in the experimental results section.

F. Framework Architectures for Image Denoising and SR



(a) Original image I^{gt} (b) Noisy image I^{lq} (c) Structure I_s^{lq} (d) Texture I_t^{lq}

Figure 6. Image "parrot" with noise level 50 and its corresponding structure and texture component.

As we described previously, for image denoising, the underlying AWGN can be regarded as an HF artifact, which can be extracted from the distorted image by the image decomposition module. However, unlike the compression artifacts, this noise is randomly overlaid on the image, and it dramatically affects the original HF information (Fig. 6(b)). Consequently, the distorted image can be decomposed into a blurred structure component (Fig. 6(c)) and a noisy texture component (Fig. 6(d)). While taking the computational efficiency into account, in this task, the very noisy texture layer is shelved, and the texture stream is removed, as are the lateral connections. Therefore, the whole framework for the denoising task can be represented by the image decomposition module leading two DL networks connecting in a cascaded way (shown in Fig.7(a)). For image SR, the dominant artifacts introduced by the inaccurate estimation of the missing pixels can be regarded as LF artifacts. Hence, the architecture of the proposed framework for image SR is even more compact than that for image denoising. Without the image decomposition module, there are only two DL networks connecting in a cascaded way. In this case, the bicubic upsampled image is fed into the quality enhancement network directly as the LF component of the distorted images and the aggregation network is utilized to further enhance the image quality (shown in Fig.7(b)). The superior performance of the proposed framework architectures will be shown in the experimental results, Sec. IV-B and Sec. IV-C, and named "Pro-DE" and "Pro-SR", respectively. The architecture summary of each framework can be found in Table I, where "ID" represents the image decomposition module; "QEN" represents the quality enhancement network; "LC" represents the lateral connections; and "AN" represents the aggregation network. Moreover, we adopt the framework "Pro-C", which has two cascaded networks QEN and AN as the single-stream benchmark framework.

Table I
Architecture summary of each framework

Framework	Pro-CAR	Pro-DE	Pro-SR	Pro-C
# of Conv layers	60	40	40	40
# of Filters/layer	128	128	128	128
Filter size	3	3	3	3
# of Para(k)	1039	594	594	594
Involved modules	ID, QEN(S&T), LC, AN	ID, QEN(S), AN	QEN, AN	QEN, AN
Target	CAR	Denoising	SR	Benchmark

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

Table II
Average PSNR(dB) and SSIM results of different methods for image CAR task on LIVE1 dataset [47].

Algorithms / Metrics	QF 10											
	JPEG	SA-DCT	ARCNN	DnCNN-3	RED30	DRRN	ARN	MemNet	MWCNN	IDN	Pro-C	Pro-CAR
PSNR	27.77	28.66	28.73	29.19	29.32	29.21	29.27	29.45	**29.69**	29.28	29.32	29.41
SSIM	0.7910	0.7977	0.8001	0.8123	0.8161	0.8146	0.8077	0.8193	**0.8254**	0.8158	0.8157	0.8191
Algorithms / Metrics	QF 20											
	JPEG	SA-DCT	ARCNN	DnCNN-3	RED30	DRRN	ARN	MemNet	MWCNN	IDN	Pro-C	Pro-CAR
PSNR	30.07	30.82	30.89	31.59	31.69	31.19	31.34	31.83	**32.04**	31.55	31.59	31.73
SSIM	0.8680	0.8658	0.8670	0.8802	0.8817	0.8678	0.8696	0.8846	**0.8885**	0.8783	0.8790	0.8824
# of Para(k)	—	—	106	665	4131	297	1145	677	16140	692	594	1039
Runtime (s)	—	18.66	0.32	—	10.65	3.88	—	2.10	—	0.51	0.99	1.64

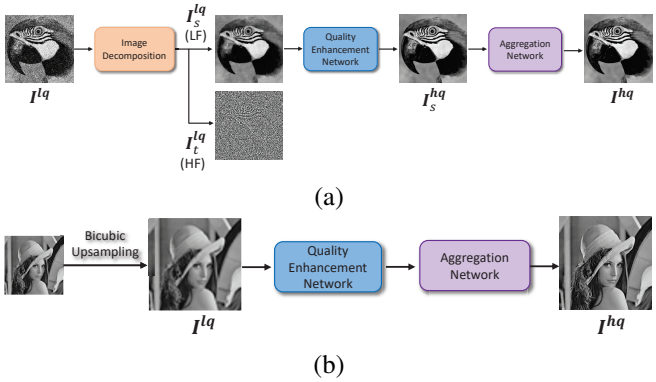


Figure 7. Corresponding frameworks for image denoising (a) and SR (b).

IV. Experimental Results

In this section, we conduct various experiments on three representative image restoration tasks: JPEG image CAR, Gaussian image denoising, and SISR. By adjusting the architecture of the proposed framework and adopting the proposed quality enhancement network, our method outperforms the state-of-the-art image restoration algorithms in terms of PSNR and SSIM [46]. In this paper, we only focus on the restoration of the luminance channel (in YCrCb space). Our implementations are available on our project webpage[1].

Taking into account the network training complexity, the small patch training strategy is adopted. Training images are split into 31×31 patches with a stride of 21. The Adam [48] solver is used with a batch size of 32 (for the dual-stream framework of image CAR) or 64 (for the cascade framework of image denoising and SR). The first and second momentum parameters are set to 0.9 and 0.999, respectively, and the weight decay is 10^{-4}. The initial learning rate is 0.001, which then decays every 5 epochs by a factor of 10. Unless otherwise specified, all the convolutional layers have 128 filters of size 3×3. We use the deep learning platform Caffe [49] on an NVIDIA GTX TITAN X GPU with 12 GB of RAM. The training time of a 40-layer network for image CAR is approximately 4 days, and for image denoising and SR, it is approximately 2 days.

A. Image Compression Artifacts Reduction

In this subsection, we first present the specific implementation details. Then, we compare the proposed framework with state-of-the-art algorithms or networks on three benchmark datasets. Subsequently, the running time of CAR approaches is discussed. Finally, we provide the ablation performance analysis of each module of the proposed CAR framework.

1) Implementation Details: In this experiment, we apply the standard JPEG compression scheme, and use the JPEG quality settings QF 20 (mid-quality) and QF 10 (low-quality) in the MATLAB JPEG encoder. For choosing the value of λ_s in this task, we follow the suggestion in [12], such that for higher QF, smaller λ_s needs to be chosen, and $0.02 \leq \lambda_s \leq 0.05$. Hence, in our work, at QF 10, $\lambda_s = 0.04$. The MSE value of the training images at QF 10 is approximately double that at QF 20, i.e., $\frac{MSE(\mathbf{I}^{QF10})}{MSE(\mathbf{I}^{QF20})} \approx 2$; therefore, the value of λ_s for QF 20 is set to half of that of QF 10, i.e., $\lambda_s = 0.02$. To yield the texture layers from the GT images for training, the value of λ_{gt} is set to 0.02.

For a fair comparison, we follow [6] and adopt 400 images from the BSDS500 database [50] as the training set. For validation, we adopt the widely used datasets Set14 [51], LIVE1 [47], BSDS500 [50] (the remaining 100 images) and DIV2K [52]. BSDS500 and DIV2K are large-scale datasets; in particularly, each image in DIV2K has a resolution up to 2K and is full of details. Hence, it is a challenging dataset for image restoration. Data augmentation is performed: first, we rotate the training images by $90°$, $180°$, and $270°$; and second, we flip them horizontally. Consequently, with a patch size of 31×31, we extract $985,600$ training image pairs.

Table III
Average PSNR(dB)/SSIM results of different methods for image CAR task on BSDS500 [50] and DIV2K [52].

Methods		BSDS500 QF 10	BSDS500 QF 20		DIV2K QF 10	DIV2K QF 20
JPEG		26.62/0.7690	29.71/0.8325		29.54/0.8183	32.04/0.8802
SA-DCT		28.38/0.7678	30.45/0.8432		30.66/0.8481	33.03/0.8958
ARCNN		28.46/0.7702	30.46/0.8441		30.94/0.8546	33.32/0.9009
RED30		28.93/0.7862	31.11/0.8586		31.25/0.8605	33.73/0.9073
DRRN		28.87/0.7850	30.72/0.8430		31.20/0.8588	33.71/0.9065
MemNet		**29.02/0.7890**	**31.23/0.8613**		31.45/0.8644	33.91/0.9099
IDN		28.87/0.7844	30.98/0.8534		31.22/0.8598	33.61/0.9045
Pro-C		28.92/0.7838	30.72/0.8434		31.19/0.8593	33.79/0.9072
Pro-CAR		28.99/0.7877	31.15/0.8577		**31.58/0.8779**	**34.17/0.9275**

2) Quantitative Comparisons: In this part, the proposed dual-stream framework for image CAR is quantitatively compared with 9 state-of-the-art approaches, including the non-DL-based method: SA-DCT [53], and DL-based methods: ARCNN [6], RED30 [8], DRRN [21], DnCNN-3 [7], ARN

[1]The source code is available at https://github.com/jzrita/Flexible_Deep_CNN_for_IR.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

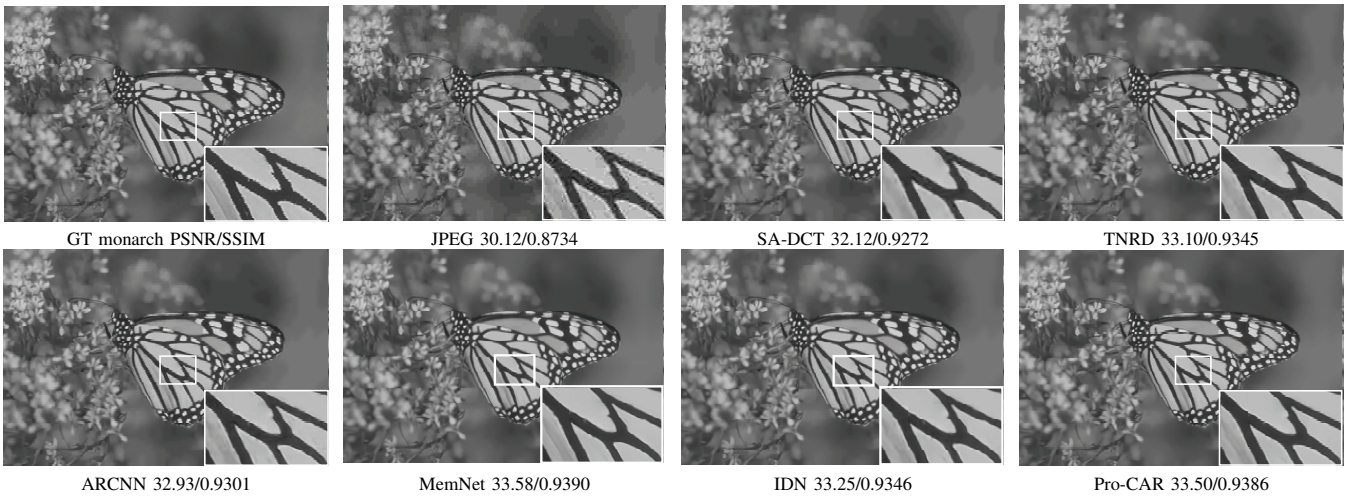
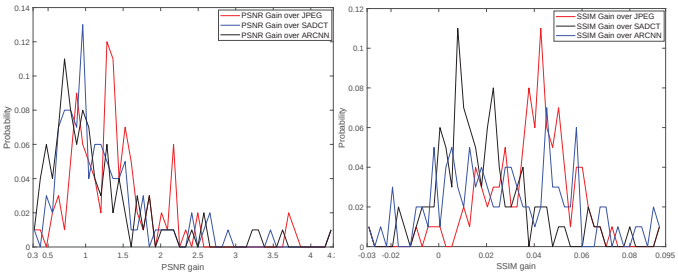


Figure 8. Visual quality comparison for image "monarch" from Set14 [51] for image CAR task at QF 10. The corresponding PSNR(dB) and SSIM results are listed below the corresponding images. This figure is best viewed on screen.

[9], MemNet [10], MWCNN [11] and IDN [41] [2]. Table II and Table III show the quantitative results on LIVE1, BSDS500 and DIV2K datasets with JPEG quality (QF) 10 and 20, respectively. In Table II, the proposed dual-stream framework with comparatively less training parameters can still achieve the performance ranking in the top three on the LIVE1 dataset. Compared with the retrained ARCNN with zero paddings that has only 4 convolutional layers, our deep architecture obtains a 0.68 dB and 0.84 dB PSNR gain at QF 10 and 20, respectively, and the improvement in terms of SSIM can be up to 0.0190 and 0.0154. Even larger PSNR gains can be obtained on BSDS500 and DIV2K reported in Table III. This comparison demonstrates the benefit of the very deep architecture. Moreover, compared with RED30 which is a single-stream network, with 78.41% less training parameters, the proposed framework can still achieve a 0.09 dB and 0.04 dB PSNR gain at QF 10 and 20 on LIVE1, respectively. This comparison demonstrates the effectiveness of the proposed dual-stream architecture and the recursive learning strategy. Compared with MWCNN, which has the best performance on LIVE1 dataset, the proposed framework saves 93.56% of GPU memory usage for storing the model. Compared with MemNet, which has the second best performance, the proposed framework has a faster running speed. Therefore, the proposed framework has a better tradeoff between model size, running time and performance. Moreover, by achieving the best results on the challenging dataset DIV2K, the proposed framework shows its superiority in reconstructing the images with full textures.

In Fig. 9, we draw the probability distribution of PSNR and SSIM gains over several baselines for CAR on BSDS500 at QF 10. Apparently, our gains over all these baselines in both PSNR and SSIM are positive in the vast majority of the test images, which shows the superiority of the proposed method.

3) Qualitative Comparisons: Qualitative comparison results for QF 10 are shown in Fig. 8. Strong blocking artifacts are visible in the JPEG compressed images. SA-DCT produces



(a) (b)

Figure 9. The distributions of the PSNR gains and SSIM gains of the proposed method over baselines for the image CAR task on the BSDS500 dataset at QF 10.

blurring effects on image edges. ARCNN significantly improves the image quality; however, it also oversmooths some image details. TNRD recovers the compressed images well. However, MemNet, IDN, and Pro-CAR reconstruct the images with sharper edges than TNRD. The deeper the network is, the sharper the edges are.

4) Running Time: In addition to visual quality, another important aspect for an image restoration method is the running speed. For a fair comparison, we profile the time consumption of all the non-DL-based algorithms in a MATLAB 2015b environment on a PC with an Intel CPU at 3.30 GHz with 16 GB RAM. While all the DL-based algorithms are implemented using the same GPU environment[3]. Table II shows the average running time[4] of different algorithms for image CAR at QF 10 on LIVE1. It can be seen that as a deep network, the proposed 40-layer dual-stream framework can still achieve lower time consumption than the networks with fewer layers. For example, DRRN has 20-layers with a network depth that is half of ours but has consumed 2.24s more time on each image. With a simpler

[2]When we conduct IDN on CAR and denoising tasks, the deconvolutional layer has been replaced by a convolutional layer.

[3]DnCNN and MWCNN are implemented in MatConvNet; however, we failed to set up the same testing environment. Hence, there is no running time for them.

[4]We run each algorithm on LIVE1 5 times, and each time calculate the average time consumption on each image. Then, the final average running time is obtained by averaging the 5 running times.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

architecture than Pro-CAR, the Pro-C consumes less time than the dual-stream architecture. However, this inevitably suffers from some performance decrease.

5) Ablation Analysis: In this subsection, we evaluate the effectiveness of each module in the proposed Pro-CAR by comparing corresponding frameworks without the particular modules. Table IV shows the ablation analysis results. In Table IV, the compared frameworks from top-to-bottom are: our baseline DnCNN, which has none of the proposed modules; a dual-stream framework with image decomposition (ID) but using DnCNN instead of the quality enhancement network (QEN); a deep QEN with 6 recursive blocks (RBs) where each RB has 3 residual units (RUs); the single-stream framework Pro-C; Pro-CAR without the lateral connections (LC) and the aggregation network (AN), which can also be regarded as "dual-QEN"; Pro-CAR without LC; Pro-CAR without AN; and our proposed framework Pro-CAR.

Table IV
Ablation Analysis of each module for image CAR task on LIVE1 at QF10

Framework	ID	QEN	LC	AN	PSNR
DnCNN	×	×	×	×	29.19
Dual-DnCNN	✓	×	×	×	28.49
Deep QEN	×	✓	×	×	29.34
Pro-C	×	✓	×	✓	29.32
Pro-CAR w/o LC&AN	✓	✓	×	×	28.77
Pro-CAR w/o LC	✓	✓	×	✓	29.38
Pro-CAR w/o AN	✓	✓	✓	×	28.79
Pro-CAR	✓	✓	✓	✓	**29.41**

By comparing "Dual-DnCNN" with baseline DnCNN, we find that although the artifacts are reduced separately, without AN, the simple addition of two streams' outputs cannot well complement each other. Benefiting from the multipath recursive residual learning that contributes to the flow of information and the gradient, "Deep QEN" improves the performance of the baseline. While comparing "Pro-C" with "Deep QEN", we find that both frameworks are composed of a single-stream architecture; however, one consists of two cascaded networks and the other is one deep network. Since they have similar network depth, their performance is also similar. To evaluate the effectiveness of LC and AN, we conduct another three experiments by removing one or both of them from Pro-CAR. The last four rows in Table IV show that these new frameworks suffer from performance degradation when removing LC or AN. When removing AN, the total network depth decreases from 40 to 20, hence, its performance decreases significantly. Through these quantitative analyses, the effectiveness and benefits of our proposed ID, QEN, LC, and AN are well demonstrated.

6) Extension on JPEG2000: JPEG2000 coding is based on the discrete wavelet transform, which generally introduces blurring and ringing compression artifacts. In this subsection, the effectiveness of the proposed method in working with JPEG2000-compressed images is tested and compared with ARCNN, DRRN, MemNet, and ERP-CA [56] baselines. All the DL networks are retrained on the training images that are compressed using the JPEG2000 encoder from MATLAB software at 0.1 bit per pixel (BPP). The involved λ_s and λ_{gt} for image decomposition are 0.04 and 0.02, respectively. The performance is presented in Table V, which is measured by the average PSNR and SSIM over the testing set. Apparently, by reducing the artifacts according to their characteristics (i.e., HF or LF artifacts), the proposed framework surpasses all other baselines. Moreover, the networks designed for deblocking or relying on DCT domain prior may fail in this case. Therefore, adopting the dual-stream to separately suppress the HF and LF artifacts makes the proposed framework more robust in solving different kinds of CAR problems.

Table V
Average PSNR(dB) and SSIM results of reducing JPEG2000 artifacts on LIVE1 at quality 0.1 BPP.

Metrics	JPEG2000	EPR-CA	ARCNN	DRRN	MemNet	Pro-CAR
PSNR	27.74	27.94	27.94	28.30	28.31	**28.41**
SSIM	0.7302	0.7331	0.7345	0.7459	0.7459	**0.7493**

7) Effectiveness of Recursive Learning: Fig. 10 shows the comparison between QEN with and without recursive learning when testing for the image CAR task on LIVE1 at QF 10. Due to recursive learning, the proposed network efficiently reduces compression artifacts while enjoying low storage and computational complexity demands. Moreover, while training, the recursive learning network is more stable and obtains better results than the nonrecursive one. Referring to Fig. 10, both versions trained from scratch achieve a better performance than that of JPEG after the first 4 epochs, and they outperform ARCNN after approximately 6 epochs.

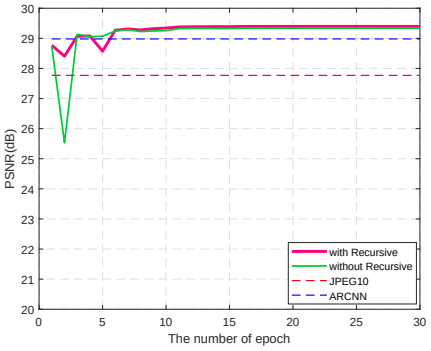


Figure 10. The comparison between the proposed network with and without recursive learning when testing for the image CAR task on LIVE1 at QF 10.

B. Image Denoising

In this subsection, we first present the specific implementation details for the image denoising and then compare the proposed framework for denoising with state-of-the-art approaches on the widely used benchmark datasets. Finally, we discuss the running time of the denoising approaches.

1) Implementation Details: In this experiment, we enhance the image quality of distorted images suffering from AWGN at specific noise levels, i.e., σ = 15, 25 and 50. The value of λ_s for noise level 50 is determined by exhaustive search within the range $[0.0001\ 0.1]$. Then, similar to the CAR experiments, based on the MSE ratio between the MSE of training images at noise level 50 and the other two levels, corresponding λ_s

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

Table VI
Average PSNR(dB)/SSIM results of different methods for image denoising task on the BSD68 [57] dataset

Methods	BM3D	TNRD	DnCNN-3	MemNet	MWCNN	IDN	Pro-C	Pro-DE
σ=15	31.08/0.8722	31.42/0.8826	31.46/0.8826	31.44/0.8793	31.86/0.8947	32.22/0.9006	31.54/0.8846	**32.40/0.9131**
σ=25	28.57/0.8017	28.92/0.8157	29.02/0.8190	29.14/0.8220	29.41/0.8360	29.92/0.8502	30.64/0.8787	**31.02/0.8904**
σ=50	25.62/0.6869	25.97/0.7029	26.10/0.7076	26.73/0.7397	26.53/0.7366	27.57/0.7741	28.31/0.8186	**28.53/0.8229**
# of Para(k)	0	0	665	677	16140	862	594	594
Runtimes(s)	1.26	1.88	—	0.54	—	0.15	0.40	0.40

values are determined. Hence, the λ_s values at noise level 15, 25 and 50 are 0.002, 0.005 and 0.02, respectively. In this task, the GT images are the original noise-free images.

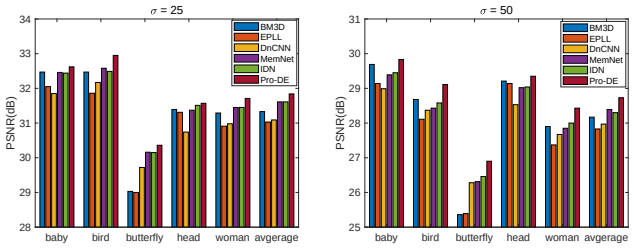
For fair comparison, we follow [28] and [7] that use a dataset that consists of 91 images from [47] and 200 training images from the Berkeley segmentation dataset BSDS500 [50] as our training set. For validation, we adopt the BSD68 [57] dataset containing 68 natural images, the Set 5 and the DIV2K dataset [52]. Note that all these testing images are widely used for the evaluation of Gaussian denoising methods, and they are not included in the training dataset. Moreover, in this task, data augmentation is performed as well. By adopting the same patch size of 31×31, we extract $575,552$ training image patch pairs.

2) Quantitative Comparisons: Compared with 7 state-of-the-art denoising methods, including BM3D [33], EPLL [58], TNRD [31], DnCNN [7], MemNet [10], MWCNN [11] and IDN [41], the corresponding results on dataset BSD68 are shown in Table VI. The proposed framework for denoising is named as "Pro-DE" in this table. From these results, the proposed approach shows significant superiority on both PSNR and SSIM results compared with the remaining state-of-the-art approaches. In particular, the gain compared to that of the second best approach reaches 0.18 dB, 0.38 dB, and 0.96 dB at σ= 15, σ= 25 and σ= 50, respectively. According to [61], few methods can outperform BM3D by more than 0.3 dB on average. In contrast, the proposed approach outperforms BM3D by 2.23 dB on average for the three noise levels. Improved SSIM performance can also be observed. This comparison demonstrates that it is beneficial to reduce the artifacts based on their characteristics.

Table VII
Average PSNR(dB) and SSIM results of different methods for image denoising task on the DIV2K [52] dataset

Methods	σ=15		σ=25		σ=50	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EPLL	33.44	0.9036	31.04	0.8531	27.82	0.7540
DnCNN-3	33.45	0.9042	31.04	0.8592	27.89	0.7702
MemNet	33.72	0.9009	30.52	0.8340	27.50	0.7471
IDN	33.46	0.8976	30.36	0.8280	27.01	0.7499
Pro-C	33.88	0.9088	31.33	0.8574	28.16	0.7887
Pro-DE	**34.25**	**0.9224**	**31.84**	**0.8804**	**28.76**	**0.7949**

Table VII lists the average PSNR and SSIM results for the different methods when applied on DIV2K dataset. It can be seen that even on such a challenging dataset the proposed approach can still maintain its advantages over the other state-of-the-art methods at all noise levels. To be specific, the proposed Pro-DE exceeds the second best method by 0.37 dB, 0.51 dB and 0.60 dB at σ= 15, σ= 25 and σ= 50, respectively.



(a) (b)

Figure 11. Denoising PSNR results on each image in Set5 with noise level (a) 25 and (b) 50.

The performance comparison on each image in the Set5 [62] dataset at noise level 25 and 50 is shown in Fig. 11. It is clear that the proposed approach has superior performance over the other approaches on each image.

3) Qualitative Comparisons: Fig. 12 shows two visual quality comparisons for the test images "pepper" and "parrot" at noise level 25 and 50, respectively. In general, BM3D is prone to oversmooth the images, and this becomes more obvious when the noise level increases. During the learning process, TNRD and DnCNN try to preserve sharp edges and fine details; however, they are likely to generate artifacts in the smooth region (see the "parrot" image). In contrast, the proposed approach, even at noise level 50, can still recover the fine details while removing noise.

4) Running Time: In this subsection, the running time of BM3D and TNRD are evaluated on the same PC as previously. We run each algorithm 5 times for noise level σ=15 and report the average execution time in Table VI. As a learning-based method, since TNRD is implemented in a parallel computing way, it costs little time. In contrast, benefiting from weights sharing, our Pro-DE reconstructs noise-free images at a fast speed.

C. Image Super-resolution

In this subsection, we first present the specific implementation details for the image SR task and then compare the proposed framework for image SR with state-of-the-art approaches on the widely used benchmark datasets.

1) Implementation Details: In this task, the image decomposition module is not employed in the framework; therefore, no parameter selection is required. To train the network, we follow [28] and [7] and adopt the same training dataset as used for the denoising task. For validation, we adopt the widely used datasets BSDS500 [50], Urban100 [63] and DIV2K [52]. We employ the same data augmentation as before. Consequently, with a patch size of 31×31, we extract $575,552$ training image patch pairs. Similar to [5], we first downsample

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

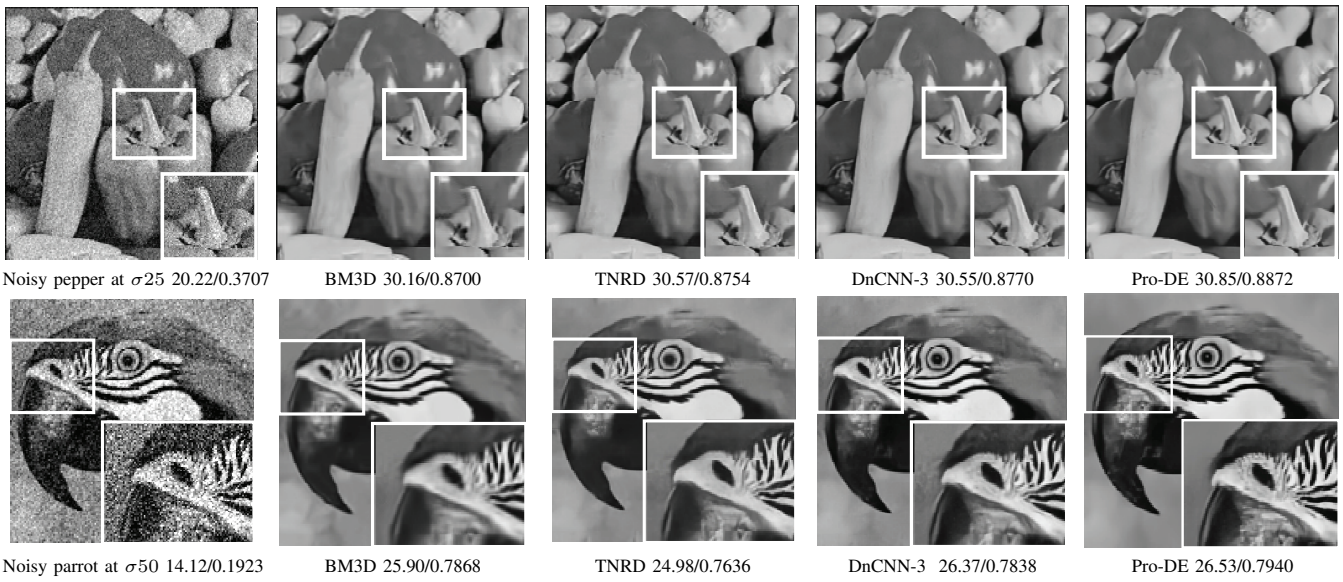


Figure 12. Denoising visual quality comparison for images "pepper" at σ=25 and "parrot" at σ=50. The corresponding PSNR(dB) and SSIM results are listed below the images. This figure is best viewed on screen.

Table VIII
Average PSNR(dB)/SSIM results of different methods for image SR task on three commonly used large-scale datasets

Dataset	Upscale	Bicubic	SRCNN	VDSR	DnCNN-3	ARN	MemNet	MWCNN	IDN	Pro-SR
BSDS500	2	29.10/0.8267	30.14/0.8610	31.89/0.8961	31.90/0.8961	30.44/0.8624	32.08/0.8978	**32.23/0.8999**	32.08/0.8985	32.03/0.8980
	3	27.01/0.7308	27.92/0.7700	28.82/0.7980	28.85/0.7981	28.02/0.7732	28.96/0.8001	**29.12/0.8060**	28.95/0.8013	28.87/0.7993
	4	25.82/0.6625	26.53/0.6957	27.28/0.7256	27.29/0.7253	26.69/0.7036	27.40/0.7281	**27.62/0.7355**	27.41/0.7297	27.33/0.7274
Urban100	2	26.52/0.8242	28.20/0.8655	30.76/0.9143	30.74/0.9139	28.60/0.8750	31.31/0.9195	**32.30/0.9296**	31.27/0.9196	30.70/0.9200
	3	24.30/0.7258	25.57/0.7756	27.13/0.8283	27.15/0.8276	25.85/0.7898	27.56/0.8376	**28.13/0.8514**	27.42/0.8359	26.56/0.8185
	4	23.03/0.6509	23.98/0.6965	25.17/0.7528	25.20/0.7521	24.24/0.7128	25.50/0.7630	**26.27/0.7890**	25.41/0.7632	25.22/0.7546
DIV2K	2	32.43/0.9041	34.40/0.9315	33.43/0.9250	35.42/0.9400	—/—	**35.62/0.9416**	—/—	35.00/0.9308	35.61/0.9410
	3	29.65/0.86306	30.95/0.8616	30.77/0.8612	31.78/0.8799	—/—	31.93/0.8799	—/—	31.92/0.8728	**32.19/0.8818**
	4	28.11/0.7748	29.10/0.8030	29.08/0.8041	29.84/0.8231	—/—	29.97/0.8268	—/—	29.90/0.8111	**30.11/0.8290**
# of Para(k)		—	8	665	665	1145	677	16140	862	594
Runtimes(s)		—	3.21	0.34	—	—	0.56	—	0.22	0.40

each image in the training set by using the bicubic algorithm of MATLAB with scale factors of 2, 3 and 4. Then, we train the models respectively for different scale factors.

2) Quantitative Comparisons: In this subsection, bicubic upsampling is used as the basic benchmark and we compare the performance of the proposed Pro-SR with another 8 learning-based approaches: SRCNN [5], TNRD [31], VDSR [28], DnCNN [7], ARN [9], MemNet [10], MWCNN [11] and IDN [41] on datasets Set5 [62], Set14 [51], BSDS500 [50] and Urban100 [63]. The BSDS500 dataset consists of natural scenes, and the Urban100 set contains challenging images of urban scenes with significant texture details. It is sufficient to indicate the performance of these approaches by evaluating them on these commonly used datasets. The corresponding PSNR and SSIM results are reported in Table VIII. The proposed framework outperforms existing methods SRCNN, TNRD, VDSR, DnCNN, and ARN in most cases. On the BSDS500 dataset, the proposed framework outperforms the well-known network DnCNN, with the improvement margin of 0.13 dB, 0.02 dB, and 0.04 dB on scale factors 2, 3 and 4, respectively. Even with fewer training parameters, the proposed framework can still achieve very competitive results compared to those of the best network on the difficult Urban100 dataset.

3) Qualitative Comparisons: The qualitative comparisons of image "ppt3" at scale factor 3 and image "Urban068" at scale factor 4 are provided in Fig. 13. At scale 3, the capability of SRCNN and VDSR to suppress the LF blurring artifacts is inadequate, i.e., the edges remain blurred. DnCNN performs well at small scale factors but cannot fully recover sharp edges at large scale factors. The proposed framework produces the best visual quality with fewer unpleasant artifacts and sharper reconstructed edges, even at the large scale factor.

4) Running Time: In this subsection, the running time of each method in Table VIII is obtained by testing on BSDS500 at scale 2. Since the source code provided by SRCNN implements the testing process in the CPU, the running time is longer than other methods while employing a shallow network depth. Comprehensively comparing Pro-SR with the state-of-the-art methods from the performance, memory cost and running time perspectives, the proposed framework achieves a better tradeoff situation.

D. Framework Convergence

In this subsection, we examine the convergence characteristic of the proposed framework on three IR tasks by evaluating the PSNR evolution in the epochs. The PSNR evolutions of the corresponding datasets on three IR tasks versus the number

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2938340, IEEE Transactions on Multimedia

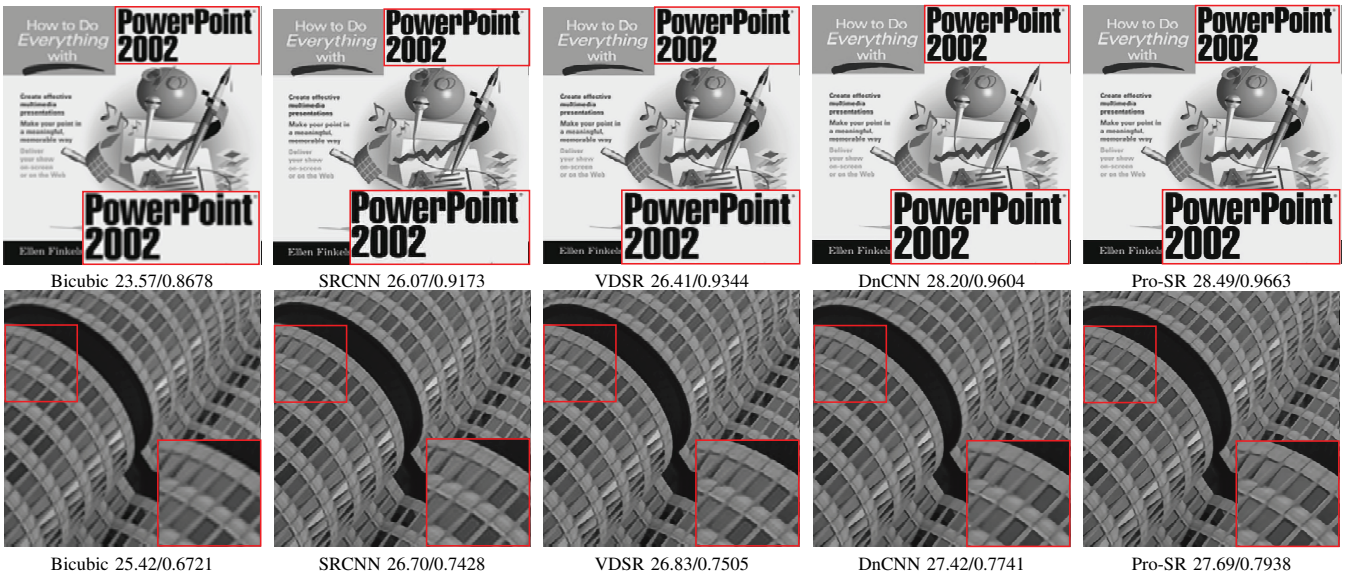


Figure 13. Image SR visual quality comparison for images "ppt3" from Set14 [51] and "Urban068" from [63] at scale factors 3 and 4, respectively. The corresponding PSNR(dB) and SSIM results are listed below the images. This figure is best viewed on screen.

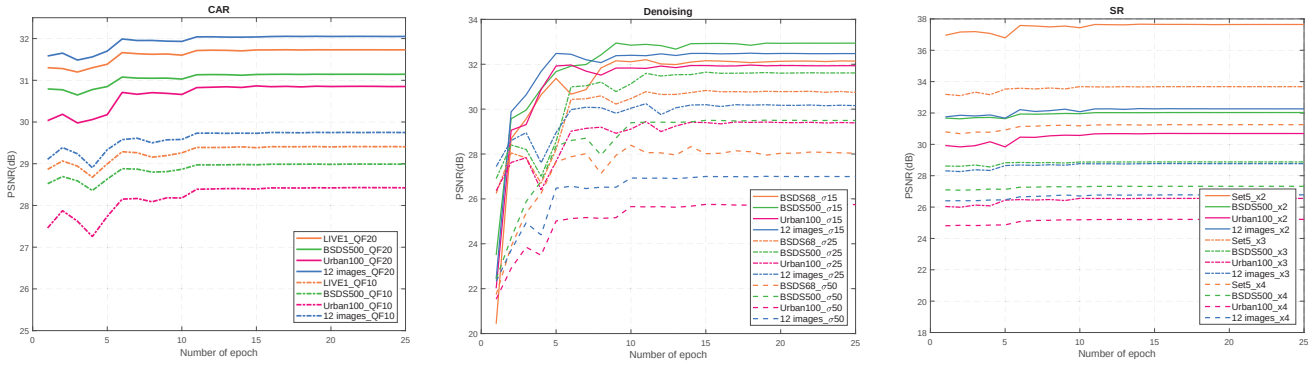


Figure 14. The convergence characteristic of the proposed frameworks for different tasks.

of epochs are shown in Fig. 14. At the early training stage, all the curves monotonically increase in general, which provides an indication of the effectiveness of the proposed frameworks. As the epoch number increases, all the curves converge to a certain level. It appears that 10 epochs are sufficient to achieve convergence.

V. Conclusion

In this work, we propose a flexible deep CNN framework for image restoration, which exploits the frequency characteristics of different types of artifacts. For specific IR tasks, the artifacts are first decomposed into a high-frequency or low-frequency group based on their characteristics. Then, according to the decomposition outcomes, the proposed framework can be efficiently adjusted to suppress these artifacts separately by the proposed quality enhancement network. In the proposed network, the multipath design helps the gradient flow and the transmission of the low-level features. Residual learning eases the training process. Recursive learning allows reducing the number of training parameters. In this way, the proposed flexible framework can significantly handle different artifacts while requiring fewer training parameters and less running time than the state-of-the-art approaches. Our future work will address an extension of the proposed framework from the quality enhancement of still images to video sequences by exploring the temporal relations.

References

[1] C. Ren, X. He, and T. Q. Nguyen, "Adjusted non-local regression and directional smoothness for image restoration," *IEEE Transactions on Multimedia*, pp. 1–1, 2018.

[2] T. Li, X. He, L. Qing, Q. Teng, and H. Chen, "An iterative framework of cascaded deblocking and superresolution for compressed images," *IEEE Transactions on Multimedia*, vol. 20, no. 6, pp. 1305–1320, Jun. 2018.

[3] J. Yin, B. Chen, and Y. Li, "Highly accurate image reconstruction for multimodal noise suppression using semisupervised learning on big data," *IEEE Transactions on Multimedia*, pp. 1–1, 2018.

[4] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, Jun. 2017.

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

[6] C. Dong, Y. Deng, C. Change Loy, and X.Tang, "Compression artifacts reduction by a deep convolutional network," in *ICCV*, IEEE, Dec. 2015, pp. 576–584.

[7] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[8] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *NIPS*, Dec. 2016, pp. 2802–2810.

[9] Y. Zhang, L. Sun, C. Yan, X. Ji, and Q. Dai, "Adaptive residual networks for high-quality image restoration," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3150–3163, Mar. 2018.

[10] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A Persistent Memory Network for Image Restoration," in *ICCV*. IEEE, Oct. 2017, pp. 4539–4547.

[11] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-Level Wavelet-CNN for Image Restoration," in *CVPR Workshops*. IEEE, Sep. 2018, pp. 174–188.

[12] Y. Li, F. Guo, R.T. Tan, and M.S. Brown, "A contrast enhancement framework with jpeg artifacts suppression," in *ECCV*. Springer, Sep. 2014, pp. 174–188.

[13] P. Korus, J. Biaas, and A. Dziech, "Towards practical self-embedding for jpeg-compressed digital images," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 157–170, Feb. 2015.

[14] L. Kang, C. Hsu, B. Zhuang, C. Lin, and C. Yeh, "Learning-based joint super-resolution and deblocking for a highly compressed image," *IEEE Transactions on Multimedia*, vol. 17, no. 7, pp. 921–934, Jul. 2015.

[15] D.-A. Huang, L.-W.. Kang, Y.-C. Wang, and C.-W. Lin, "Self-learning based image decomposition with applications to single image denoising," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 83–93, Jan. 2013.

[16] L. Kang, C. Lin, and Y. Fu, "Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742-1755, Apr. 2012.

[17] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944-2956, Jun. 2017.

[18] S. Gnther, L. Ruthotto, J.B. Schroder, E.C. Cyr, N.R. Gauger "Layer-Parallel Training of Deep Residual Neural Networks," *arXiv preprint arXiv:1812.04352*, 2018.

[19] S. Yan, C. Wu, L. Wang, F. Xu, L. An, K. Guo, and Y. Liu; , 2018, pp. 151-167 "DDRNet: Depth Map Denoising and Refinement for Consumer Depth Cameras Using Cascaded CNNs," *ECCV*, Springer, 2018, pp. 151-167.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, IEEE, Jun. 2016, pp. 770–778.

[21] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *CVPR*, IEEE, Jul. 2017.

[22] H. Liu, R. Xiong, J. Zhang, and W. Gao, "Image denoising via adaptive soft-thresholding based on non-local samples," in *CVPR*, IEEE, Jun. 2015, pp. 484–492.

[23] W. Wang, C. Ren, X. He, H. Chen and L. Qing, "Video Super-Resolution via Residual Learning," *IEEE Access*, vol. 6, pp. 23767– 23777, Apr. 2018.

[24] M. Xu, S. Li, J. Lu, and W. Zhu, "Compressibility constrained sparse representation with learnt dictionary for low bit-rate image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1743–1757, Oct. 2014.

[25] H. Wang, X. Gao, K. Zhang, and J. Li, "Single image super-resolution using gaussian process regression with dictionary-based sampling and student-t likelihood," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3556–3568, Dec. 2017.

[26] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4613–4626, Dec. 2013.

[27] J. Liu, S. Yang, Y. Fang, and Z. Guo, "Structure-guided image inpainting using homography transformation," *IEEE Transactions on Multimedia*, pp. 1–1, 2018.

[28] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *CVPR*, IEEE, Jun. 2016, pp. 1646–1654.

[29] L. Cavigelli, P. Hager, and L. Benini, "Cas-cnn: A deep convolutional neural network for image compression artifact suppression," in *IJCNN*, IEEE, May 2017, pp. 752–759.

[30] J. Guo and H. Chao, "Building dual-domain representations for compression artifacts reduction," in *ECCV*. Springer, Oct. 2016, pp. 628–644.

[31] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[32] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *CVPR*, IEEE, Jun. 2012, pp. 2392–2399.

[33] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, Jul. 2007.

[34] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *NIPS*, Dec. 2012, pp. 341–349.

[35] Y. Shi, K. Wang, C. Chen, L. Xu, and L. Lin, "Structure-preserving image super-resolution via contextualized multitask learning," *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2804–2815, Dec. 2017.

[36] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *CVPR*, IEEE, Jun. 2016, pp. 1637–1645.

[37] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *CVPR*, IEEE, Jun. 2017, pp. 624–632.

[38] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *CVPR Workshops*, IEEE, Jun. 2017, pp. 4.

[39] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *CVPR*, IEEE, Jun. 2018.

[40] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *ECCV*. Springer, Sep. 2018, pp. 8–14.

[41] Z. Hui, X. Wang, and X. Gao, "Fast and Accurate Single Image Super-Resolution via Information Distillation Network," in *CVPR*, IEEE, Jun. 2018.

[42] L. Xu, Q. Yan, Y Xia and J. Jia, "Structure Extraction from Texture via Relative Total Variation," *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 1–10, Nov. 2012.

[43] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, Jul. 2009.

[44] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 248–272, Jul. 2008.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *ECCV*. Springer, Oct. 2016, pp. 630–645.

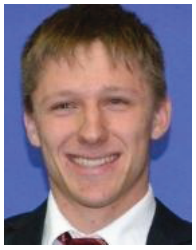
[46] Z. Wang, AC Bovik, HR Sheikh, and EP Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[47] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on*

Image Processing, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
[48] MD. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.
[49] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACMMM*, ACM, Nov. 2014, pp. 675–678.
[50] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, IEEE, Jul. 2001, vol. 2, pp. 416–423.
[51] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *ICCS*, Springer, Jun. 2010, pp. 711–730.
[52] R. Timofte, S. Gu, J. Wu, L. Van Gool, L. Zhang, M.-H. Yang, M. Haris, and others, "NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results," *CVPR Workshops*, IEEE, Jun. 2018.
[53] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, May 2007.
[54] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4613–4626, Dec. 2013.
[55] H. Chang, M. K. Ng, and T. Zeng, "Reducing artifacts in jpeg decompression via a learned dictionary," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 718–728, Feb. 2014.
[56] R. Rothe, R. Timofte, and L. Van, "Efficient regression priors for reducing image compression artifacts," in *ICIP*, IEEE, Sep. 2015, pp. 1543–1547.
[57] S. Roth and M. J. Black, "Fields of experts: a framework for learning image priors," in *CVPR*, IEEE, Jun. 2005, pp. 860–867.
[58] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *ICCV*, IEEE, Nov. 2011, pp. 479–486.
[59] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *CVPR*, IEEE, Jun. 2014, pp. 2862–2869.
[60] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *CVPR*, IEEE, Jun. 2014, pp. 2774–2781.
[61] A. Levin, B. Nadler, F. Durand, and W.T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *ECCV*, Springer, Oct. 2012, pp. 73–86.
[62] C. M. Bevilacqua, A. Roumy, and M. Morel, "Low complexity single-image super-resolution based on nonnegative neighbor embedding," in *BMVC*, BMVA Press, Sep. 2012, pp. 1–10.
[63] J. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *CVPR*, IEEE, Jun. 2015, pp. 5197–5206.

Zhi Jin (M'16-current) received the B.S degree in Telecommunication Engineering from the University of Liverpool (UoL), UK and Xi'an Jiaotong-Liverpool University (XJTLU), P.R. China, in 2011. She received the Ph.D. degree from the University of Liverpool, UK in 2016. From 2016-2018, she worked as a Postdoctoral Researcher in Shenzhen University, while during 2017-2018, she jointly worked in Technical University of Munich (TUM) as a research fellow. Her current research interests include 2D/3D image/video quality enhancement, and 3D reconstruction.

Muhammad Zafar Iqbal received the BE and MS in Electrical Engineering from Balochistan University of Engineering and Technology, Kuzdar, Pakistan and National University of Sciences and Technology, Islamabad, Pakistan, in 2008 and 2013, respectively. Currently, he is pursuing a PhD degree from Technical University of Munich, Germany. In 2016, he joined the Chair of Media Technology at Technical University of Munich, Germany as a PhD scholar. His PhD work is supported by a scholarship granted by the Higher Education Commission (HEC) of Pakistan in collaboration with DAAD Germany. His research interests include 2D and 3D image processing, computer vision, deep learning, artificial intelligence, and robotics.



Dmytro Bobkov studied electrical engineering at the Technical University of Munich (Germany) and National Technical University of Ukraine (Ukraine). From 2013 to 2018, he was a member of research staff of the Chair of Media Technology, Technical University of Munich, Germany, where he was working towards his PhD degree in the area of 3D computer vision and machine learning. His current research interests lie in the area of multi-view computer vision and machine learning.

Wenbin Zou received the M.E. degree in software engineering with a specialization in multimedia technology from Peking University, P.R. China, in 2010, and the Ph.D. degree from the National Institute of Applied Sciences, Rennes, France, in 2014. From 2014 to 2015, he was a Researcher with the UMR Laboratoire dinformatique Gaspard-Monge, CNRS, and Ecole des Ponts ParisTech, France. Since then, he has been with the College of Electronics and Information Engineering, Shenzhen University, P.R. China. His current research interests include saliency detection, object segmentation, and semantic segmentation.

Eckehard Steinbach (M'96-SM'08-F'15) studied electrical engineering at the University of Karlsruhe, Germany, the University of Essex, U.K., and ESIEE Paris, received the Ph.D. degree in engineering from the University of Erlangen-Nuremberg, Germany, in 1999. From 1994 to 2000, he was a Member of the Research Staff of the Image Communication Group at the University of Erlangen-Nuremberg. From 2000 to 2001, he was a Post-Doctoral Fellow with the Information Systems Laboratory, Stanford University. In 2002, he joined the Department of Electrical Engineering and Information Technology, Technical University of Munich, Germany, where he is currently a Full Professor of media technology. His current research interests are in the area of audio-visual-haptic information processing and communication, and networked and interactive multimedia systems.

Xia Li received her B.S. and M.S. in electronic engineering and SIP (signal and information processing) from Xidian University in 1989 and 1992 respectively. She was later conferred a Ph.D. in Department of information engineering by the Chinese University of Hong Kong in 1997. Since then, she has been with the College of Electronics and Information Engineering, Shenzhen University, P.R. China. Her research interests include intelligent computing and its applications, image processing and pattern recognition.