This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

Color Image Restoration Exploiting Inter-channel Correlation with a 3-stage CNN

Kai Cui, *Student Member, IEEE*, Atanas Boev, Elena Alshina, and Eckehard Steinbach, *Fellow, IEEE*

***Abstract*—Image restoration is a critical component of image processing pipelines and for low-level computer vision tasks. Conventional image restoration approaches are mostly based on hand-crafted image priors. The inter-channel correlation of color images is not fully exploited. Motivated by the special characteristics of the inter-channel correlation (higher correlation for red/green and green/blue channels than for red/blue) in color images and general characteristics (green channel always shows the best image quality among the three color components) of distorted color images, in this paper, a 3-stage convolutional neural network (CNN) structure is proposed for color image restoration tasks. Since the green channel is found to have the best quality among all three channels, in the first stage, the network is designed to reconstruct the green component. Then, with the guidance of the reconstructed green channel from the first stage, the red and blue channels are reconstructed in the second stage with two parallel networks. Finally, the intermediate reconstructions from the previous stages are concatenated and further refined jointly. We demonstrate the capabilities of the proposed 3-stage structure with three typical color image restoration tasks: color image demosaicking, color compression artifacts reduction, and real-world color image denoising. In addition, we integrate pixel-shuffle convolution into our scheme to improve the efficiency, and also introduce a quality-blind training strategy to simplify the training process for the compression artifacts reduction task. Extensive experimental results and analyses show that the proposed structure successfully exploits the spatial and inter-channel correlation of color images and outperforms the state-of-the-art image reconstruction approaches.**

***Index Terms*—Color image restoration, Inter-channel Correlation, Convolutional Neural Network, Demosaicking, Compression artifacts reduction, Realistic image denoising**

I. Introduction

IMAGE restoration (IR) is a fundamental task in image processing, which aims at reconstructing high-quality images from distorted observations. Color images suffer from different kinds of distortions introduced during image acquisition or along the processing pipeline, including missing samples, sensor noise, compression, and transmission distortion. In order to solve these issues, many image restoration approaches have been proposed in the past years. Conventional image restoration approaches are mostly based on image priors [1]. The IR problems are formulated as iterative optimization problems and regularized by image priors, including low-rank, sparsity, patch-wise smoothness, and non-local similarity [2]–[4]. These image priors have clear physical meaning, and they are general features that can handle different IR problems. However, most of them are hand-crafted and not optimal for a specific task. Also, solving an iterative optimization problem is usually time-consuming.

Recently, with the success of convolutional neural networks (CNN) and deep learning (DL) in image processing, many learning-based image restoration approaches have been proposed (e.g. [5]–[10]). These approaches are based on data-driven end-to-end learning, and most of them outperform conventional approaches for various IR tasks.

However, no matter for conventional algorithms or learning-based approaches, most of them are focusing only on spatial correlation exploration. The algorithms are usually only performed on gray-scale images or the luminance component of color images. Although some approaches [6], [10]–[14] are extended to deal with color image restoration (CIR) tasks, the inter-channel correlation is still not well exploited. These approaches directly apply the algorithm on different color channels separately, adopt hand-designed pipeline or simply expand the input number of channels, which does not leverage the statistics of different color channels.

In this paper, inspired by the special characteristics of the inter-channel correlation among red (R), green (G), blue (B) channels in color images, and the general characteristics of distorted color images, a 3-stage CNN structure is proposed for CIR tasks. For distorted color images, we have observed that the G channel usually has the best quality among the three channels. In the first stage, the network is designed to reconstruct the G channel. Then, in the second stage, due to the strong but different inter-channel correlations between the R/G and the G/B channels, two parallel networks are designed to restore R and B channels with the guidance of the high-quality G values reconstructed in the first stage. Finally, the intermediate R, G, B reconstructed from the previous stages are concatenated and further refined jointly in the third stage.

A preliminary version of the 3-stage CNN scheme was presented in our previous work [15] for color image demosaicking and in [16] for compression artifacts reduction. They outperform the related works and achieve the state-of-the-art performance for the corresponding CIR tasks. In addition to our previous works, the main contributions of the proposed approach in this paper are:

(1) A detailed analysis of the inter-channel correlation of natural color images which indicates that the R/G and G/B channels exhibit much stronger correlations than R/B. Additionally, the analysis shows that the correlations for R/G and G/B have different characteristics. By theoretical and experimental analyses of the statistics for typical distorted color images, we prove that the G channel has the best quality among all three channels, and that this is a general feature of

K. Cui and E. Steinbach are with the Chair of Media Technology, Technical University of Munich, Munich, 80333, Germany (e-mail: kai.cui@tum.de, eckehard.steinbach@tum.de).

A. Boev and E. Alshina are with Huawei Technologies Duesseldorf GmbH, Munich, 80992, Germany (e-mail: atanas.boev@huawei.com, elena.alshina@huawei.com).

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

various types of distorted color images.

(2) Inspired by the specific inter-channel correlation and the distorted color image characteristics, a 3-stage CNN scheme is proposed for the CIR tasks, in which the inter-channel correlation is fully exploited to improve the quality of the reconstructed images. Compared to the preliminary version in [15], [16], we modify the network structure to make it more compact and efficient. For this, the pixel-shuffle scheme [17] is integrated into the 3-stage CNN, which not only solves the initialization issue for the demosaicking task but also further improves the efficiency and performance.

(3) Along with the proposed 3-stage CNN scheme, a quality-blind training strategy is introduced for the color compression artifacts reduction task. It makes the training process much more manageable and allows a single trained network to handle the images compressed with a wide-range of distortion levels. To the best of our knowledge, this is the first work that shows the suitability of a single trained model to cover the whole quality factor (QF) range of JPEG encoders and which brings consistent quality improvements for any JPEG QF value.

(4) Extensive evaluation, comparison with existing approaches, and comprehensive analyses on multiple CIR tasks and datasets have been performed, which prove the superiority, efficiency, and effectiveness of the proposed scheme.

The remainder of this paper is organized as follows. Section II discusses the related works. Section III presents a detailed analysis of natural color images, including inter-channel correlation and the characteristics of different kinds of distorted color images. Section IV explains the details of our proposed 3-stage CNN-based CIR scheme. Section V presents the experimental details, results, and analysis. Section VI analyzes the computational complexity, the effectiveness of the 3-stage structure, the pixel-shuffle layer, and the quality-blind training strategy. Finally, Section VII concludes the paper.

II. Related Work

In this paper, we solve CIR problems with CNNs and DL. In particular, we focus on three important CIR tasks, including color image demosaicking (CDM), color compression artifacts reduction (CAR), and real-world color image denoising (RIDN). In the remainder of this section, we first discuss general CIR approaches, and then mainly focus on the existing CNN and DL-related approaches that are targeting these three problems.

A. Color Image Restoration

Conventional CIR approaches are usually inspired by the corresponding gray-scale IR algorithms. CBM3D proposed in [11] is one of the examples. The color images are first decorrelated into a luminance-chrominance color space by a hand-crafted linear transform. Then the gray-scale BM3D [18] method is applied to each transformed channel. Color image priors are also often adopted in CIR approaches, and these priors are usually the extension of the gray-scale image priors. In [12], [19], [20], sparse color image representation approaches were proposed. By extending the K-SVD from gray-scale images to color images, these proposed approaches learn the correlation between R/G/B channels. They achieve better performance compared to modeling each channel separately. In [14], the color non-local prior was introduced to CIR tasks and outperforms other local regularization methods. Color total variation minimization models were proposed in [13] and [21] for CIR tasks. The gray-scale total variation (TV) is extended to the color TV and the inter-channel vectorial color TV. The color TV regularization brings performance improvements for various types of CIR tasks. For DL-based CIR algorithms, most approaches [6], [10], [22] directly expand the input of the channels to three and increase the depth of the network. The correlation of different color channels is explored implicitly in the network through supervised training.

B. Color Image Demosaicking

CDM is a typical CIR task in which the distortion is introduced by missing samples. In Bayer pattern-based [23] cameras, half of the green and three-fourths of the red and blue samples are missing. Bayer pattern color filter array (CFA) based CDM has been extensively studied in the past years. Many image priors have been introduced to reconstruct the full-color images from the mosaicked CFA images [24]. Recently, CNN-based CDM algorithms have been proposed. In [25], a deep joint demosaicking and denoising structure was proposed, which can solve these two problems with a single trained network. A residual denoising network was proposed in [26] to solve the same joint problem, which adopts an iterative network structure to demosaick and enhance CFA images distorted by a wide range of noise levels. In [27], a 2-stage network was proposed, which reconstructs the G channel in the first stage and further refines the R/G/B channels jointly in the second stage. In [28], a multiple-model fusion-based network was proposed. A general model, a rough texture, and a smooth texture model are adopted to generate multiple estimations of the image. Then these estimations are weighted and fused to generate the final demosaicked image. These CNN-based approaches achieve much better performance than conventional algorithms and represent the state-of-the-art performance.

C. Compression Artifacts Reduction

CAR effectively improves the visual quality of compressed images without increasing the bit-rate. Traditional blockwise transform-based image codecs, especially JPEG, suffer from strong blocking and ringing artifacts. Conventional approaches design filters based on the position of block boundary and image priors [1]. Recently, CNN-based CAR algorithms have been proposed. ARCNN was proposed in [29], which is a simple structure of only four CNN layers. It achieves significant performance improvements compared to conventional approaches. Residual learning and batch normalization based network structure DnCNN was proposed in [6] to solve multiple image restoration problems, including denoising, super-resolution and JPEG deblocking with a single trained network. Recently, many CNN-based approaches are focusing on transform domain enhancement and multi-scale feature extraction and fusion [30], [31]. In [32], the compressed images

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

are converted to the frequency domain using the multi-scale discrete wavelet transform (DWT), and all the convolution operations are performed in the wavelet domain. A dual-domain compression algorithm artifact reduction approach called IDCN was proposed in [10]. With the pixel position labeling map, quantization table, and quality factor (QF) as side information, the network explores the priors in the discrete cosine transform (DCT) domain. However, most of these approaches were proposed only for gray-scale images, and the inter-channel correlation is neglected.

D. Real-world Color Image Denoising

Conventional denoising algorithms assume that the noise in an image is additive white Gaussian noise (AWGN). The noise in different regions of the images and the color channels are assumed to be independent and to have the same noise levels. However, these algorithms usually fail in the RIDN task because realistic noise is more complex than AWGN and has quite different distributions [33]–[35]. In order to solve these issues, a multi-channel optimization model was proposed in [34]. The realistic denoising task is formulated as a weighted nuclear norm minimization problem, and a weight matrix is introduced to balance the data fidelity of the three channels in consideration of their different noise statistics. In [35], a trilateral weighted sparse coding scheme was proposed to characterize the statistics of realistic noise. Recently, more real-world noisy image datasets with a large number of images became publicly available [36], [37], and data-driven learning-based approaches were proposed. In [38], a convolutional blind denoising network was proposed. The network consists of a noise estimation subnetwork to estimate the noise levels and a U-Net like structure to reconstruct the clean images. In [39], a residual learning and feature attention based denoising network was proposed, in which the feature attention is adopted to exploit the channel dependencies. These two CNN-based approaches achieve much better performance than other approaches for many RIDN benchmark datasets.

III. Inter-channel Correlation and Distorted Color Image Characteristics Analysis

A. Inter-channel Correlation Analysis

Since exploiting the inter-channel correlation is a key point for CIR tasks, we first analyze its characteristics. The inter-channel correlation of a color image can be explained from both the structural and spectral points of view. From the structural aspect, the samples of the three color channels share similar texture, structures, and edges of the objects, which leads to strong dependencies and correlation among the three channels.

From the spectral point of view, the captured wavelengths of R, G, and B are quite close to each other, and often have significant overlap. In [40]–[42], the spectral sensitivity functions of various types of color camera sensors are analyzed. Fig. 1 shows the typical spectral sensitivities of each color channel for a digital color camera sensor. The spectral sensitivity curves show that the sensors capture the visible color spectrum with the wavelength range 400nm–700nm. The sensitivity curves of the three color channels are overlapping. This means that the reconstruction of the other two color channels can significantly benefit from the samples which are measured for the current channel.

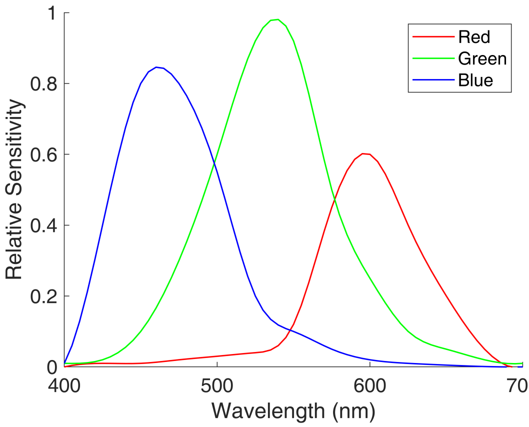


Fig. 1. The typical spectral sensitivities of a modern digital RGB camera sensor. The figure is adapted from Fig. 6 in [40].

However, the correlation of R/G, G/B, and R/B shows different characteristics. The wavelength differences of R/G and G/B are much smaller than those of R/B, and the three channels usually show quite different intensity value distributions. As shown in Fig. 1, the spectral sensitivity curves of R/G and G/B have large overlapping areas, and the sensitivity amplitude in the overlapping areas is high. The R/B curves have a much smaller overlapping area, and the sensitivity amplitude in that overlapping area is much lower.

In order to illustrate these characteristics of the inter-channel correlation quantitatively, several publicly available benchmark datasets are adopted to evaluate the inter-channel correlation coefficients between R/G, G/B, and R/B, respectively. The Kodak [43] and McMaster [44] datasets are adopted because these two are the most commonly used CIR benchmark datasets. Kodak HD is the original high-resolution version of the Kodak dataset. These three datasets have a very limited number of images and scenes, which may lead to an undesired bias when evaluating the correlation coefficients. Therefore, we also adopt high-quality image datasets with a larger number of natural images, including the DIV2K [45], the WED [46], and the RAISE dataset [47]. The number of images in each dataset is marked in Tab. I.

The Pearson correlation coefficient is employed to measure the inter-channel correlation, which is shown in Equation (1).

$$r = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(A_{ij}-\bar{A})(B_{ij}-\bar{B})}{\sqrt{(\sum_{i=1}^{M}\sum_{j=1}^{N}(A_{ij}-\bar{A})^2)(\sum_{i=1}^{M}\sum_{j=1}^{N}(B_{ij}-\bar{B})^2)}} \tag{1}$$

$$\bar{A} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}A_{ij}, \bar{B} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}B_{ij}$$

where A and B are intensity values for the considered color components of an image, $\bar{A}$ and $\bar{B}$ are the corresponding mean

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

TABLE I
MEAN AND VARIANCE OF INTER-CHANNEL CORRELATION COEFFICIENTS FOR VARIOUS IMAGE DATASETS

Correlation	Channels	Kodak (24)	Kodak HD (24)	McMaster (18)	DIV2K (800)	WED (4744)	RAISE (8156)
Mean	R/G	0.8722	0.7866	0.7445	0.8985	0.9010	0.9401
	G/B	0.9056	0.9083	0.7114	0.9036	0.9149	0.8970
	R/B	0.7669	0.7086	0.5618	0.7590	0.7892	0.8021
Variance	R/G	0.0172	0.0529	0.0606	0.0180	0.0149	0.0082
	G/B	0.0133	0.0090	0.0887	0.0160	0.0105	0.0166
	R/B	0.0365	0.0642	0.0841	0.0623	0.0407	0.0397

values of these color components, and M and N are the width and height of the image.

Tab. I shows the mean and variance of the correlation coefficients between R/G, G/B, and R/B for these datasets. From the results, it can be seen that, first, for all the datasets, the correlation values of the R/G and G/B channels are consistently higher than those of the R/B channels (higher mean values and lower variance values of the correlation coefficients). This is in line with the observation mentioned above that the wavelength differences of R/G and G/B are smaller than those of R/B.

We also observe that for R/G and G/B, both of them have high correlations, but the characteristics of the correlation are quite different. They have very similar mean values but quite different variance values, which means that the correlation coefficients have quite different distributions. As the Kodak dataset is a downsampled version of the Kodak HD dataset, the results for these two datasets show that the resolution also affects the correlation characteristics. The loss of high-frequency details caused by downsampling leads to higher inter-channel correlation. The correlation values of the McMaster dataset are abnormally low compared to the other datasets, which may be caused by the very small number of images (18 images) and their unique characteristics. The DIV2K, WED, and RAISE datasets contain large amounts of images, and the results better represent the general case of natural images. All three datasets show very typical characteristics as we analyzed before. In summary, the three channels have strong pairwise correlations, but the correlation between every two channels is quite different.

B. Characteristics of Distorted Color Images

In our experiments with raw, compressed, and real-world noisy images, in almost all cases, the G channel shows the best image quality among the three channels. In Bayer pattern raw images, the G channel has twice as many samples than the R and B channels. Apparently, for the corresponding demosaicked images, the G channel has much better reconstruction quality compared to the other two channels. For compressed and real-world noisy color images, the explanation is not that straightforward. In the remainder of this section, we will analyze the characteristics of compressed and real-world noisy color images in detail.

We take the JPEG image codec as an example for analyzing the characteristics of compressed color images. JPEG has a simple structure, and it is the most commonly used image compression standard. Many other advanced image compression algorithms are based on the blockwise transform compression scheme proposed in JPEG. JPEG compression is an $8*8$ blockwise DCT based image compression scheme, and the major encoding and decoding steps are shown in Fig. 2. These steps include color transform, DCT, quantization, entropy encoding, and the corresponding inverse processing steps. The distortion of the compressed image is majorly introduced by the quantization step Q. Other steps are lossless or approximately lossless considering rounding errors.

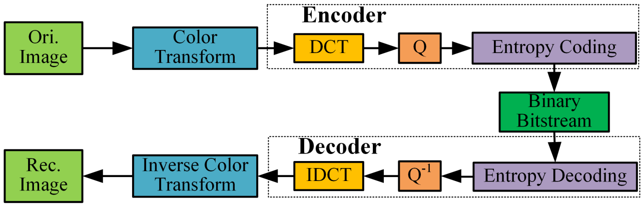


Fig. 2. Blockwise transform-based color image compression scheme.

In image and video compression, the YCbCr color space is usually used instead of RGB. The YUV420 color format is the most commonly adopted color format, in which the chrominance components are downsampled by a factor of 2 horizontally and vertically. This is based on the assumption that the human visual system is less sensitive to color differences compared to brightness changes. The uncompressed original image in RGB color space is $\mathbf{I_{RGB}}$ and it has three color components $\mathbf{I_R}$, $\mathbf{I_G}$ and $\mathbf{I_B}$. The color space conversion matrix and the corresponding inverse conversion matrix of the JPEG standard [48] are shown in Equation (2) and (3).

$$\begin{bmatrix}\mathbf{I_Y}\\ \mathbf{I_{Cb}}\\ \mathbf{I_{Cr}}\end{bmatrix} = \begin{bmatrix}0.299 & 0.587 & 0.114\\ -0.169 & -0.331 & 0.500\\ 0.500 & -0.419 & -0.081\end{bmatrix}\begin{bmatrix}\mathbf{I_R}\\ \mathbf{I_G}\\ \mathbf{I_B}\end{bmatrix} + \begin{bmatrix}0\\ 128\\ 128\end{bmatrix} \quad (2)$$

$$\begin{bmatrix}\mathbf{I_R}\\ \mathbf{I_G}\\ \mathbf{I_B}\end{bmatrix} = \begin{bmatrix}1 & 0 & 1.402\\ 1 & -0.344 & -0.714\\ 1 & 1.772 & 0\end{bmatrix}\begin{bmatrix}\mathbf{I_Y}-0\\ \mathbf{I_{Cb}}-128\\ \mathbf{I_{Cr}}-128\end{bmatrix} \quad (3)$$

After color space conversion, the image $\mathbf{I_{YCbCr}}$ has the three channels $\mathbf{I_Y}$, $\mathbf{I_{Cb}}$ and $\mathbf{I_{Cr}}$, which represent luminance, blue and red chrominance, respectively. Then, if defining the image compression process as $\mathrm{CODEC}(\cdot)$, which includes Encoder and Decoder shown in Fig. 2, the reconstructed image in YCbCr color space is:

$$\mathbf{I'_{YCbCr}} = \mathrm{CODEC}(\mathbf{I_{YCbCr}}) \quad (4)$$

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

The Y, Cb, and Cr channels have different statistics. The luminance channel contains many high-frequency details, while the chrominance channels are mainly composed of low-frequency color difference signals. Considering these statistics and the human vision system, different compression parameters and different quantization matrices for the luminance channel and the chrominance channels are adopted in CODEC$(\cdot)$. Defining the compression process of $\mathbf{I_Y}$, $\mathbf{I_{Cb}}$ and $\mathbf{I_{Cr}}$ as $\text{CODEC}_\mathbf{Y}(\cdot)$, $\text{CODEC}_\mathbf{Cb}(\cdot)$ and $\text{CODEC}_\mathbf{Cr}(\cdot)$, respectively, then the compression error in the YCbCr color space $\Delta\mathbf{I_{YCbCr}}$ is expressed as:

$$\begin{aligned}\Delta\mathbf{I_{YCbCr}} =& \mathbf{I'_{YCbCr}} - \mathbf{I_{YCbCr}} \\ =& \begin{cases} \Delta\mathbf{I_Y} &= \text{CODEC}_\mathbf{Y}(\mathbf{I_Y}) - \mathbf{I_Y} \\ \Delta\mathbf{I_{Cb}} &= \text{CODEC}_\mathbf{Cb}(\mathbf{I_{Cb}}) - \mathbf{I_{Cb}} \\ \Delta\mathbf{I_{Cr}} &= \text{CODEC}_\mathbf{Cr}(\mathbf{I_{Cr}}) - \mathbf{I_{Cr}} \end{cases}\end{aligned} \tag{5}$$

where $\Delta\mathbf{I_Y}$, $\Delta\mathbf{I_{Cb}}$, and $\Delta\mathbf{I_{Cr}}$ are the compression errors for Y, Cb and Cr, respectively.

The compression loss is mainly introduced by the quantization process of a codec. In the JPEG compression standard, a base quantization matrix $\mathbf{Q_b}$ is defined, and different compression qualities are achieved by adjusting the quantization coefficient (QC). Therefore, the quantization matrix for a specific quality $\mathbf{Q}$ is defined as:

$$\mathbf{Q} = \left[\frac{50 + \text{QC} * \mathbf{Q_b}}{100}\right] \tag{6}$$

where $[\cdot]$ is the rounding operation. For simplicity, in practice, an integer quality factor ($\text{QF} \in [0\,..\,100]$) is defined to adjust the compression levels. The mapping relationship between QF and QC is defined in Equation (7).

$$\text{QC} = \begin{cases} \dfrac{5000}{\text{QF}}, & \text{QF} \in [0\,..\,50] \\ 200 - 2 * \text{QF}, & \text{QF} \in [51\,..\,100] \end{cases} \tag{7}$$

As analyzed in [49], after the inverse quantization step Q^{-1}, the errors between the reconstructed DCT coefficients and the true values can be formulated as a uniform distribution with a range of $[-0.5, 0.5) * \mathbf{Q}$. Since the DCT is an orthogonal transform, according to the Lindeberg-Feller Central Limit Theorem, when performing the inverse DCT (IDCT) for these quantized DCT coefficients, the compression error in the YCbCr color space $\Delta\mathbf{I_{YCbCr}}$ is modeled as an approximate Gaussian distribution as follows:

$$\begin{cases} \Delta\mathbf{I_Y} &\sim \mathcal{N}(\mu_\mathbf{Y}, \sigma^2_\mathbf{Y}) \\ \Delta\mathbf{I_{Cb}} &\sim \mathcal{N}(\mu_\mathbf{Cb}, \sigma^2_\mathbf{Cb}) \\ \Delta\mathbf{I_{Cr}} &\sim \mathcal{N}(\mu_\mathbf{Cr}, \sigma^2_\mathbf{Cr}) \end{cases} \tag{8}$$

where $\mathcal{N}(\cdot)$ is the Gaussian distribution, $\mu_\mathbf{Y}$, $\mu_\mathbf{Cb}$, $\mu_\mathbf{Cr}$ and $\sigma^2_\mathbf{Y}$, $\sigma^2_\mathbf{Cb}$, $\sigma^2_\mathbf{Cr}$ are the expectations and variances of $\Delta\mathbf{I_Y}$, $\Delta\mathbf{I_{Cb}}$, $\Delta\mathbf{I_{Cr}}$, respectively.

As the Y channel and Cb, Cr channels are compressed independently in the codec, we assume that the compression errors $\Delta\mathbf{I_Y}$, $\Delta\mathbf{I_{Cb}}$, and $\Delta\mathbf{I_{Cr}}$ also are approximately independent random variables. Then, when the compressed image $\mathbf{I'_{YCbCr}}$ is converted to RGB color space $\mathbf{I'_{RGB}}$ with Equation (3), the compression error in RGB domain $\Delta\mathbf{I_{RGB}}$ is represented as:

$$\begin{aligned} &\Delta\mathbf{I_{RGB}} = \mathbf{I'_{RGB}} - \mathbf{I_{RGB}} \\ &= \begin{cases} \Delta\mathbf{I_R} \sim \mathcal{N}(\mu_\mathbf{Y} + 1.402\mu_\mathbf{Cr}, \sigma^2_\mathbf{Y} + 1.402^2\sigma^2_\mathbf{Cr}) \\ \Delta\mathbf{I_G} \sim \mathcal{N}(\mu_\mathbf{Y} - 0.344\mu_\mathbf{Cb} - 0.714\mu_\mathbf{Cr}, \\ \qquad\qquad \sigma^2_\mathbf{Y} + 0.344^2\sigma^2_\mathbf{Cb} + 0.714^2\sigma^2_\mathbf{Cr}) \\ \Delta\mathbf{I_B} \sim \mathcal{N}(\mu_\mathbf{Y} + 1.772\mu_\mathbf{Cb}, \sigma^2_\mathbf{Y} + 1.772^2\sigma^2_\mathbf{Cb}) \end{cases} \end{aligned} \tag{9}$$

where $\Delta\mathbf{I_R}$, $\Delta\mathbf{I_G}$, and $\Delta\mathbf{I_B}$ are the compression error for R, G and B, respectively.

As Cb and Cr are both color difference signals, these two channels share the same compression parameters in the codec. We assume hence that $\mu_\mathbf{Cb} \approx \mu_\mathbf{Cr}$, $\sigma^2_\mathbf{Cb} \approx \sigma^2_\mathbf{Cr}$. Then Equation (9) can be simplified as:

$$\begin{cases} \Delta\mathbf{I_R} &\sim \mathcal{N}(\mu_\mathbf{Y} + 1.402\mu_\mathbf{Cb}, \sigma^2_\mathbf{Y} + 1.966\sigma^2_\mathbf{Cb}) \\ \Delta\mathbf{I_G} &\sim \mathcal{N}(\mu_\mathbf{Y} - 1.058\mu_\mathbf{Cb}, \sigma^2_\mathbf{Y} + 0.628\sigma^2_\mathbf{Cb}) \\ \Delta\mathbf{I_B} &\sim \mathcal{N}(\mu_\mathbf{Y} + 1.772\mu_\mathbf{Cb}, \sigma^2_\mathbf{Y} + 3.140\sigma^2_\mathbf{Cb}) \end{cases} \tag{10}$$

Assuming that $\mu_\mathbf{Y}$, $\mu_\mathbf{Cb}$, $\mu_\mathbf{Cr}$ are close enough to 0, then the error levels mainly depend on the error variance. Therefore, from Equation (10), we can conclude that $\Delta\mathbf{I_B} > \Delta\mathbf{I_R} > \Delta\mathbf{I_G}$. It means that when the reconstructed RGB image $\mathbf{I'_{RGB}}$ is obtained, the G channel has the best quality, and the R and B channels have lower quality. The B channel suffers from more distortion than R.

We can also conclude from Equation (10) that this kind of characteristic is caused by the adopted color transform matrices shown in Equation (2) and (3). For images and videos of different resolutions and different codecs, the color transform matrices and the offsets adopted are slightly different, which are defined in the ITU-R BT.601 [50], ITU-R BT.709 [51], and ITU-R BT.2020 [52] standards. However, no matter which standard is adopted, similar conclusions are obtained.

In order to prove the rationality and correctness of the assumptions and simplifications adopted in the analysis, validation experiments are performed. The MATLAB built-in JPEG codec is employed as the image codec, and the LIVE1 dataset [53] is adopted as the validation dataset. There are 29 color images in the LIVE1 dataset. These images are compressed using $\text{QF} \in [0\,..\,100]$ with a step size of 1. The average Peak Signal-to-Noise Ratio (PSNR) is adopted to evaluate the distortion of the compressed image. The rate-distortion results are shown in Fig. 3. The average PSNR values are offered for each color component separately, and for the RGB images as a whole. The 0.3–0.7 bit/pixel region is enlarged to show the details at low bit-rate.

As illustrated in Fig. 3, the G channel shows the highest PSNR values among all three channels. The R channel shows higher values than B, and the average PSNR for the RGB images are close to the performance of the R channel. Even when the bit-rate is low, as shown in the zoomed-in patch, the same characteristics can be observed. These observations are consistent with the aforementioned conclusions. Similar observations were mentioned in our previous work [16], in which the BPG image codec and the Kodak dataset were adopted to test the rate-distortion performance. This shows that

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

the assumptions adopted in the analysis are reasonable, and obtained conclusions are general characteristics of compressed color images caused by the color transform matrices.

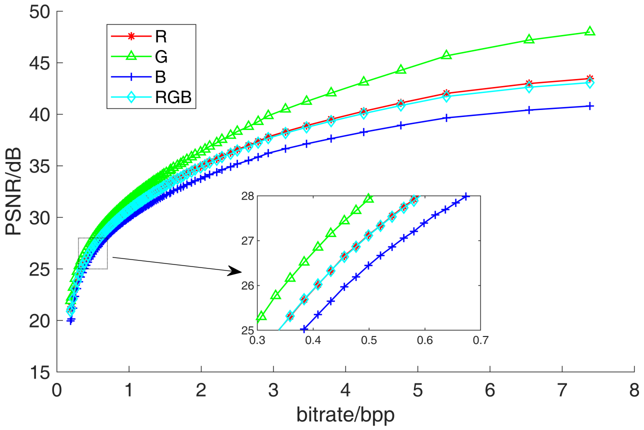


Fig. 3. PSNR of the R, G and B channels and the Composite PSNR (CPSNR) of the RGB images (JPEG, $\text{QF} \in [0\,..\,100]$, the LIVE1 dataset).

For real-world noisy images, similar characteristics are expected. Noise is introduced by the image sensor and the processing pipeline inside the camera. Most of the modern cameras are still based on the Bayer pattern CFA, in which the green channel has twice as many samples compared to the other two channels. These raw samples are distorted by noise. Then the noise is propagated and heavily mixed during the in-camera processing, including CDM, gamut mapping, tone mapping [33]. This leads to different noise levels and statistics for different color channels.

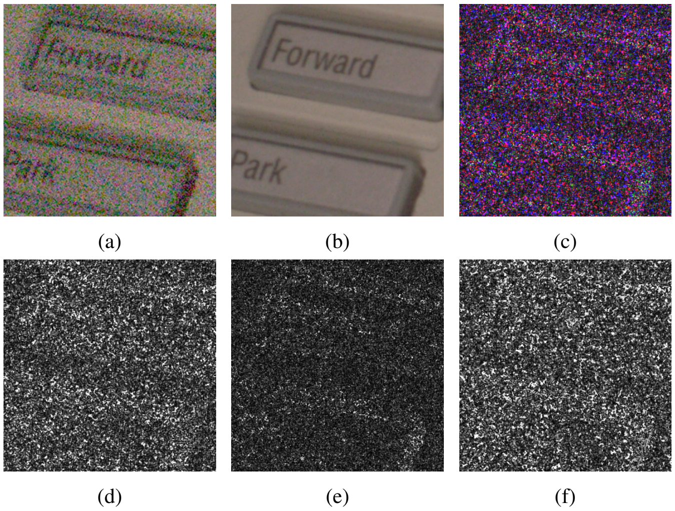

(a) (b) (c)
(d) (e) (f)

Fig. 4. Example image patch from the SIDD dataset. (a) noisy image, (b) groundtruth, (c) difference between noisy and groundtruth images, (d) difference of red channel, (e) difference of green channel, (f) difference of blue channel.

In order to better visualize these differences, real-world noisy/clean image pairs are adopted for analysis. Several real-world noisy image datasets are publicly available [33], [36], [37]. The SIDD dataset is adopted in our experiment as both the noisy images and the clean counterparts are provided. There are 320 high-resolution noisy/clean image pairs of various scenes and illumination conditions. Other datasets usually have a limited number of images and provide only the noisy images. The evaluation can only be performed with an online submission system.

First, an example image patch is shown in Fig. 4 (a). It can be seen that the image suffers from very strong noise, and the noise is spatially-variant. From the differences of each channel in Fig. 4 (d)-(e), we observe that the noise level of the green channel is much smaller compared to the other two channels.

In order to ensure that this is a general characteristic for noisy images, we evaluate the average PSNR, MS-SSIM (Multi-scale Structural Similarity Index) [54], and noise variance for the red, green, and blue channels of the whole dataset. As shown in Tab. II, the values for the green channel are about 4dB PSNR and 0.12 MS-SSIM higher compared to the red and blue channels, and the noise variance is about half compared to the other two channels. The red and blue channels have similar PSNR, MS-SSIM values, and noise variances.

TABLE II
NOISE ANALYSIS FOR THE SIDD DATASET

Channel	PSNR	MS-SSIM	Noise variance
R	26.92	0.7257	267.13
G	30.47	0.8426	133.92
B	26.70	0.7103	288.12

In summary, all these observations and analyses show that the green channel has the best quality among the three color channels and that this is a general characteristic of distorted color images. Inspired by these observations, we design a 3-stage CIR scheme, which is presented in Section IV.

IV. Proposed 3-stage CNN Scheme for Color Image Restoration

The proposed 3-stage CIR scheme is illustrated in Fig. 5. For simplicity, we take the CAR and RIDN tasks as examples to explain how the scheme works. We will explain how to adapt it to the CDM task in Section IV-A.

First, the distorted image $\mathbf{I}'_{\mathbf{RGB}}$ is processed by the pixel-shuffle layer, which rearranges each channel into four smaller image patches with quarter resolution and concatenates them as $\mathbf{X}'_{\mathbf{RGB}}$. Then, it is fed into the first stage to enhance the G channel. In the first stage, the spatial correlation and redundancies are exploited to enhance the quality of the G channel leading to $\mathbf{X}'_{\mathbf{G1}}$. Since the G channel has better quality than the other two, and R/G, G/B have very strong inter-channel correlation and the correlation usually has different characteristics, in the second stage, $\mathbf{X}'_{\mathbf{G1}}$ is adopted to guide the reconstruction of the R and B channel, respectively. By concatenating $\mathbf{X}'_{\mathbf{G1}}$ with $\mathbf{X}'_{\mathbf{R}}$ and $\mathbf{X}'_{\mathbf{B}}$ as the inputs of two parallel networks, the two subnetworks in the second stage explore the inter-channel correlation between R/G and G/B. As a result, the enhanced $\mathbf{X}'_{\mathbf{R1G2}}$ and $\mathbf{X}'_{\mathbf{G3B1}}$ are obtained. In the third stage, the intermediate R, G, B values are concatenated as $\mathbf{X}'_{\mathbf{R1G1B1}}$, fed into the third stage to further refine the quality of the image, and the 12-channel output $\hat{\mathbf{X}}_{\mathbf{RGB}}$ is obtained. Finally, in the last step, the 12-channel $\hat{\mathbf{X}}_{\mathbf{RGB}}$ is reshaped

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

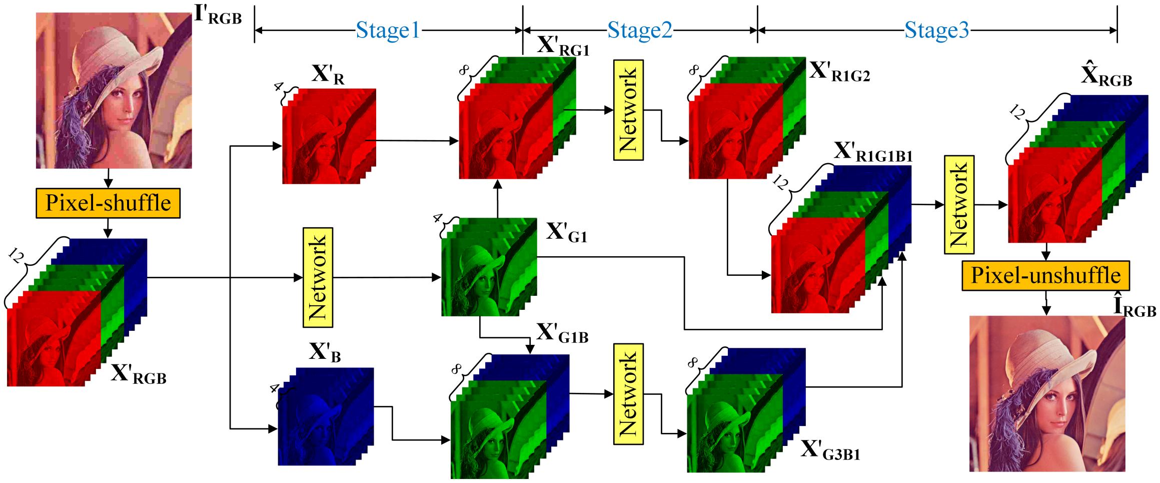


Fig. 5. Overall structure of the proposed 3-stage CNN-based color image restoration scheme.

back to the original resolution, and the enhanced RGB image $\hat{\mathbf{I}}_{\mathbf{RGB}}$ is obtained.

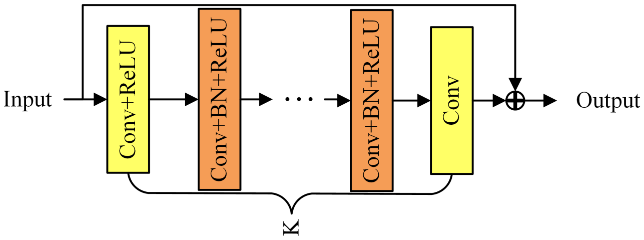


Fig. 6. Structure of the network unit for each stage (Conv: convolutional layer, ReLU: relu activation layer, BN: batch normalization layer).

Compared to our original 3-stage approach in [15] and [16], we improve the network structure to make it more compact and efficient. First, the RGB images are adopted as the input of the first stage, as the reconstruction of the G channel also benefits from the samples of the other two channels. Second, we concatenate the reconstructed G from the first stage and the R, B channels from the second stage as the input of the third stage. These modifications make sure that each stage has a clear target. The first stage is focusing on the reconstruction of the G channel. The two sub-networks in the second stage are designed to reconstruct the R and B channels, respectively. The third stage is aiming at refining the performance jointly.

Fig. 6 shows the detailed structure of the network unit for each stage. In the first layer, 64 filters of size $3\times3\times d$ are used to generate feature maps. The last convolutional layer adopts d filters of size $3\times3\times64$ to generate the corresponding output. For the hidden layers, 64 filters of size $3\times3\times64$ are adopted. The number of layers in each unit K is set to 10, and d is set to 4, 8, 12 in the three stages, respectively. Stride is set to 1, and zero-padding of size 1 is used to ensure that each feature map has the same size as the input.

A. Pixel-shuffle CNN Layer for Color Image Restoration

Pixel-shuffle convolution was initially proposed in [17] for gray-scale image super-resolution tasks. For most of the super-resolution algorithms, the input low-resolution images are first up-sampled to the desired-resolution with simple interpolation algorithms, and then adopted as the input of the super-resolution algorithms to refine the results. CDM tasks suffer from similar issues. Existing approaches mostly first interpolate the Bayer pattern image to a three-channel full-resolution image as the initialization. The result is then refined with image priors and other advanced algorithms. Our previous work in [15] also follows the same pipeline.

In [17], the authors showed that the interpolation + refinement structure is sub-optimal and adds unnecessary computational complexity. They proposed a pixel-shuffle convolution scheme to solve these issues. The sub-pixel convolution scheme extracts features in low-resolution space. Only in the last layer, the pixels from the low-resolution feature maps are rearranged to generate the high-resolution images.

There are three significant advantages of the pixel-shuffle convolution scheme. The first one is that the initialization is not required anymore. The second is the lower computational complexity, as only $1/N^2$ convolution operations are needed compared with interpolation + refinement approaches when the network settings are the same, where N is the scale factor of the super-resolution. The third is that it has a larger receptive field for the same kernel size, as more information in the contextual area can be integrated into the reconstruction of the current pixel, which is beneficial to spatial information exploration and reconstruction quality. In [7], a similar scheme is adopted in a denoising network to boost both the running speed and performance.

Therefore, we integrate pixel-shuffle convolution into our scheme to reduce the computational complexity and improve the reconstruction performance. For the CAR and RIDN tasks, the situation is similar to color image super-resolution with $N = 2$. As shown in Fig. 5, the original distorted color image is rearranged into a 12-channel structure before feeding it to the network and reshaped back to the original dimensions after obtaining the output of the network.

For the CDM task, it is slightly different. The R and B channels are sub-sampled both vertically and horizontally by a factor of 2 compared to the desired full-resolution output.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

For the G channel, since it is sampled with a quincunx pattern, we need to split the G plane into two channels. Considering $\mathbf{X}$ is a Bayer CFA image, it is reshaped to a 4-channel RGGB images $\mathbf{X_r}$ as follows:

$$\begin{aligned}\mathbf{X}_r(i,j,0) =& \mathbf{X}(2i,2j)\\ \mathbf{X}_r(i,j,1) =& \mathbf{X}(2i,2j+1)\\ \mathbf{X}_r(i,j,2) =& \mathbf{X}(2i+1,2j)\\ \mathbf{X}_r(i,j,3) =& \mathbf{X}(2i+1,2j+1)\end{aligned} \tag{11}$$

where $i = 0, \ldots, H/2$, $j = 0, \ldots, W/2$, H and W are the height and width of the CFA image. $\mathbf{X_r}$ is fed as input to the network for demosaicking.

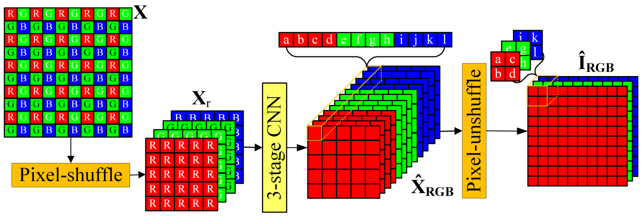


Fig. 7. Pixel-shuffle CNN for CDM (for the Bayer pattern RGGB CDM case). Illustration of how the pixel-rearrangement is performed in the first layer and in the last layer.

Fig. 7 shows the modified pixel-shuffle CNN layer for CDM and the reshaping order of the RGB image in the first and the last layer. The Bayer pattern RGGB case is illustrated here. The proposed scheme can easily be modified for other CFA patterns.

B. Quality-blind Training Strategy for CAR tasks

Most CAR approaches take JPEG compression as a benchmark to evaluate the performance. The images compressed with a specific QF, such as 10 or 20, are adopted for training, and the compressed JPEG images of the same QF are adopted for testing. In DnCNN [6], a wider range (QF $\in [10..40]$) can be covered by a single trained model, but still far from covering the whole range and only the luminance channel is considered. In IDCN [10], a limited range (QF $\in [5..20]$) can be covered. However, they need the QF values, quantization table, and the pixel position labeling map as side information. These limitations make it difficult to apply these approaches for practical CAR.

In order to solve these issues, we propose a simple but effective training strategy — quality-blind training for CAR tasks. The images in the training dataset are compressed to various distortion levels, and then these pairs of clean and compressed images are fed into the network for training. When the trained model is obtained, it has the capability to enhance the compressed color images covering the whole QF range without any side information. This strategy makes the training process much more manageable. Compared to training separately for each QF, we observe a minimal compromise on performance for our 3-stage CNN network.

For easier generation and to ensure an equal number of samples for different distortion levels, we generate the training images with QF $\in [0..100]$ and a step size of 5. Assuming there is a ground-truth RGB image $\mathbf{I}$, a sequence of 21 compressed images of different distortion levels $\{\mathbf{I}'_{\text{QF}=0}, \mathbf{I}'_{\text{QF}=5}, \cdots, \mathbf{I}'_{\text{QF}=100}\}$ is generated. In order to guarantee the trained model does not have any preference for a specific QF or type of image, these training image pairs are randomly sorted, and a training input image queue of size 5000 is created. During training, each mini-batch consists of image patches of different distortion levels randomly selected from the image input queue.

C. Loss Function

Consider the training dataset $(\mathbf{I}'_i, \mathbf{I}_i)_{i=1}^N$, where $\mathbf{I}'_i$ is the i-th distorted RGB image, $\mathbf{I}_i$ is the corresponding ground-truth RGB image, and N is the number of the images in the training data. During training, a loss function is defined to optimize the parameters of the networks. The mean squared error (MSE) function is adopted as the loss function which is defined in Equation (12).

$$\begin{aligned}&L(\omega_1, \omega_{21}, \omega_{22}, \omega_3) =\\ &\frac{1}{N}\sum_{i=1}^{N}(\|\mathcal{F}(\mathbf{I}'_i; \omega_1, \omega_{21}, \omega_{22}, \omega_3) - \mathbf{I}_i\|^2)\end{aligned} \tag{12}$$

where ω_j are the corresponding network parameters of the j-th stage. $\mathcal{F}(\mathbf{I}'_i; \omega_1, \omega_{21}, \omega_{22}, \omega_3)$ is the i-th output of the 3-stage network.

In order to ensure the generalizability of the trained model, a regularization term is adopted in the final loss function as shown in Equation (13). The regularization coefficient λ is set to 0.0005.

$$L_{all}(\omega_1, \omega_{21}, \omega_{22}, \omega_3) = L + \frac{1}{2}\lambda\sum_j(\|\omega_j\|^2) \tag{13}$$

V. Experimental Results and Analysis

The WED database is adopted as training data for the CDM and CAR tasks. In this dataset, there are 4744 high-quality natural images of various scenes. We randomly select 4644 images for the training dataset, and the remaining 100 images are used as test dataset. For the CDM task, RGGB Bayer pattern images are generated as the input of the network for training. For the CAR task, compressed images of different quality levels are generated with the MATLAB JPEG codec following the strategy introduced in Section IV-B. The SIDD dataset is adopted as the training dataset for real-world denoising tasks. It contains 320 noisy/clean image pairs for training and an additional 40 for testing.

The high-resolution images in the training datasets are cropped into small patches for training. The patch size of the input is set to 160 × 160. Random flip and rotation are adopted as the augmentation option of the training dataset. The mini-batch size is set to 64. The weights of the networks are initialized according to [55], and the Adam solver is used to optimize the parameters. The four sub-networks in the 3-stage CNN are trained end-to-end and optimized jointly. The starting learning rate is 0.001, and it is divided by ten every twenty epochs. There are 100 epochs in total. Other hyper-parameters are using the default settings from [56].

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

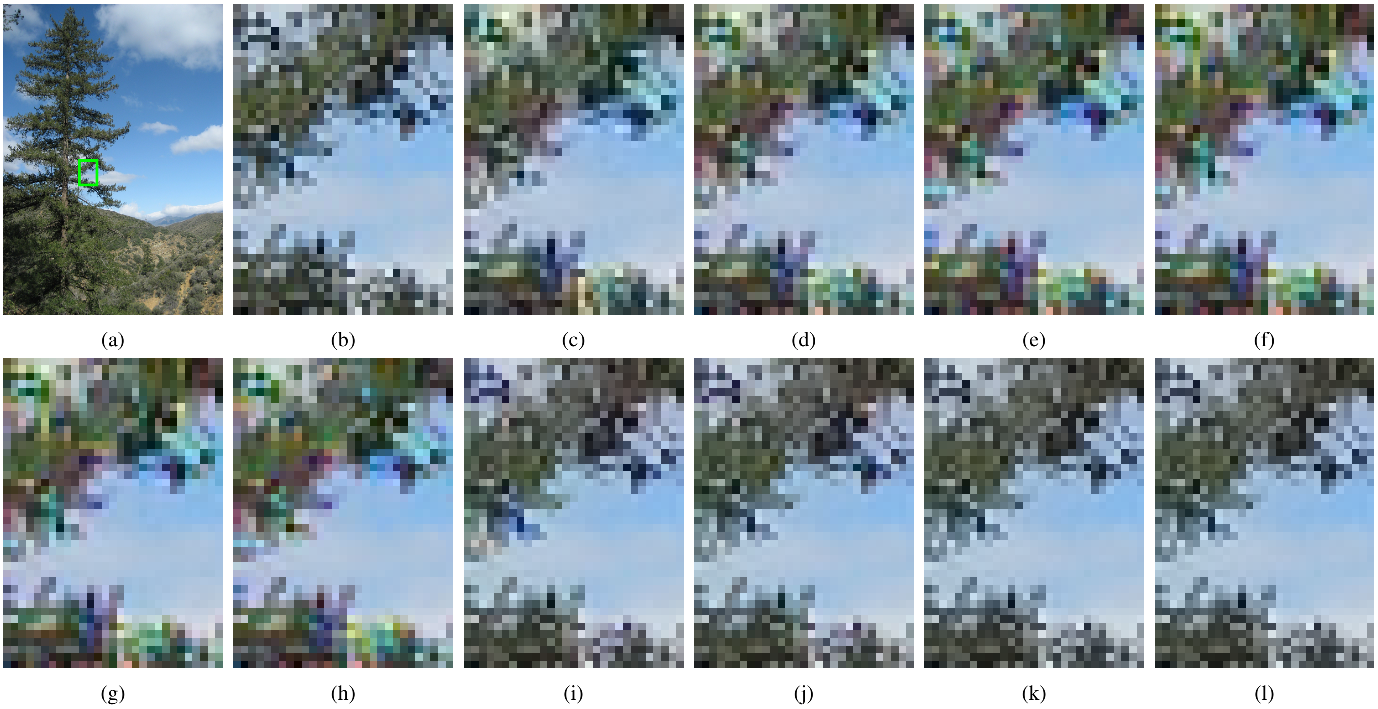


Fig. 8. Visual quality comparison on image *03168.bmp* of the WED-NEW dataset (best seen on a computer monitor). (a) Groundtruth. (b) Groundtruth Zoom-in. (c) AHD (29.01dB). (d) DLMMSE (31.09dB). (e) LDI-NAT (29.86dB). (f) RI (29.69dB). (g) MLRI (30.16dB). (h) ARI (29.98dB). (i) 2-stage (33.57dB). (j) 3-stage (33.90dB). (k) Proposed (34.62dB). (l) Proposed+ (34.73dB).

All the experiments are performed using TensorFlow 1.12, and the source code, the trained models will be available at https://github.com/amnesiack/CNNCDM3CIR.

A. Color Image Demosaicking

There are two commonly used datasets for CDM performance evaluation, the Kodak and the McMaster dataset. However, for the Kodak dataset, the images have relatively low-resolution and limited color gradations, which is not optimal for the evaluation of CDM algorithms for modern digital cameras [27]. The McMaster dataset also has limitations concerning scene variety since it contains only 18 images, and also as can be seen from Tab. I, this dataset has atypically low inter-channel correlation compared to the other natural image datasets. In order to evaluate the proposed scheme comprehensively, we also adopt the WED-NEW dataset proposed in our previous work [15] for evaluation, which is the remaining 100 images from the WED dataset.

First, an example is presented in Fig. 8 to show the visual quality of the proposed method in comparison to existing algorithms. The texture-rich and sharp color transition areas are the challenging cases for CDM. We zoom in the tree part of the image *03168.png* from the WED-NEW dataset to show the details. The results are generated with the source code provided for the reference papers without modification. *Proposed+* refers to the proposed scheme with geometric self-ensemble strategy [57].

In Fig. 8, the tree part has a very complex texture and irregular edges. With conventional approaches, many false-color artifacts can be observed, and pseudo-color pixels are generated. With the 2-stage and the original 3-stage approach, the results improve significantly, but some artifacts can still be observed along the edges of the tree. With the proposed 3-stage+pixel-shuffle approaches, these artifacts are almost eliminated, and the visual quality is further improved. Even when compared side-by-side with the groundtruth, it is difficult to see any differences.

The average PSNR and composite PSNR (CPSNR) are adopted to evaluate the objective quality of different approaches. MS-SSIM is also adopted because it is usually more robust and provides closer to human visual evaluation results. Ten pixels along the border are cropped because some algorithms suffer from border effects. The results are listed in Tab. III. The best performance of each metric is marked in red, and the second best in blue.

Tab. III reveals that compared to conventional image prior-based approaches, CNN-based approaches achieve significant PSNR improvements (usually more than 1.5dB). The proposed method leads to an additional 0.6–0.8dB PSNR improvement on different datasets in comparison to other CNN-based algorithms. The self-ensemble strategy brings about an additional 0.2dB improvement compared to the baseline approaches. The PSNR improvements for the Kodak and WED-NEW dataset are about 0.8 dB, for the McMaster dataset, it is about 0.6 dB, which is reasonable considering the inter-channel correlation in Tab. I.

B. Color Compression Artifacts Reduction

The LIVE1 and the classic5 dataset are two commonly adopted datasets for CAR tasks. However, the classic5 dataset consists of gray-scale images, which cannot be used for testing color CAR algorithms. The images in the LIVE1 dataset have

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

TABLE III
AVERAGE PSNR, CPSNR (IN DB) AND MS-SSIM RESULTS FOR THREE CDM BENCHMARK DATASETS. THE BEST PERFORMANCE IS MARKED IN RED AND THE SECOND BEST IN BLUE

Methods	Kodak (24)				McMaster (18)				WED-NEW (100)			
	R	G	B	RGB	R	G	B	RGB	R	G	B	RGB
AHD [58]	37.00	39.64	37.31	37.77/0.9947	33.00	36.98	32.16	33.49/0.9835	34.20	37.78	34.56	35.12/0.9915
DLMMSE [59]	39.18	42.63	39.58	40.11/0.9963	34.03	37.99	33.04	34.47/0.9878	35.56	39.57	35.91	36.55/0.9935
LDI-NAT [44]	37.14	39.48	37.01	37.71/0.9933	36.19	39.52	34.37	36.12/0.9922	35.62	38.69	35.71	36.37/0.9928
MLRI [60]	38.87	41.83	38.86	39.58/0.9959	36.35	39.90	35.36	36.62/0.9936	36.53	39.93	36.82	37.42/0.9945
RI [61]	38.62	41.18	38.49	39.21/0.9957	36.72	40.23	35.59	36.91/0.9941	36.49	39.64	36.76	37.32/0.9943
ARI [62]	39.27	42.43	39.10	39.95/0.9962	37.45	40.68	36.21	37.60/0.9949	36.73	40.20	36.93	37.58/0.9944
2-stage [27]	41.38	44.85	41.04	42.04/0.9974	39.14	42.10	37.31	38.98/0.9956	38.98	42.69	38.99	39.84/0.9965
3-stage [15]	42.07	45.18	41.09	42.39/0.9975	39.60	42.60	37.68	39.39/0.9958	39.32	43.04	39.37	40.19/0.9967
Proposed	42.76	45.94	41.55	42.97/0.9978	39.96	42.77	38.05	39.72/0.9960	40.04	43.60	39.98	40.84/0.9970
Proposed+	42.99	46.13	41.72	43.17/0.9978	40.22	42.93	38.22	39.91/0.9962	40.22	43.79	40.17	41.02/0.9971

relatively low resolution (768 × 512 or smaller) compared to images captured by modern cameras and smartphones. Also, it has only 29 images with limited scene diversity, which may lead to biases for some evaluation metrics. To avoid these potential issues, we adopt the image dataset proposed in the CLIC image compression challenge [63] as the evaluation dataset. The CLIC dataset includes 61 high-resolution images of daily life scenes captured with mobile devices and 41 high-quality artistic images captured by professional cameras and photographers. This dataset better represents the CAR tasks in practice.

Most of the existing CAR approaches deal with gray-scale images only and are trained for a specific quality level. In order to make a fair comparison, ARCNN [29] and DnCNN [6] are adopted and retrained for color images. The number of channels of the input layer and the output layer is modified to 3. Other parameters are the same as the original approach. These two retrained approaches are noted as ARCNN-RGB and DnCNN-RGB. 3-stage refers to the retrained version of our previous work [16] for the JPEG codec. Since these trained models can deal with multiple distortion levels with a single trained network, we select $\mathrm{QF} \in \{10, 30, 50, 70, 90\}$ to illustrate the achieved visual quality improvement. Fig. 9 provides the visual quality comparison of different algorithms. Image areas with texture, sharp edges, and color transition are the challenging cases for CAR tasks. We zoom in for the shown patches to better visualize the details.

As shown in Fig. 9, when the QF value is low ($\mathrm{QF} = \{10, 30\}$), strong compression artifacts including block, ringing, and pseudo-color pixel artifacts are observable in the compressed images. With ARCNN-RGB, these artifacts are slightly alleviated, but many visible artifacts remain. For the other three approaches, the artifacts are well removed. Among all these approaches, the proposed approach achieves the best visual quality. When the QF value is higher ($\mathrm{QF} = \{50, 70\}$), only some ringing artifacts are still visible along the sharp edges. ARCNN-RGB removes some of them, but there are still slight remaining artifacts. The other three approaches perform well, and the visual quality is improved. When the QF is very high ($\mathrm{QF} = 90$), the compressed images and the enhanced ones are visually similar.

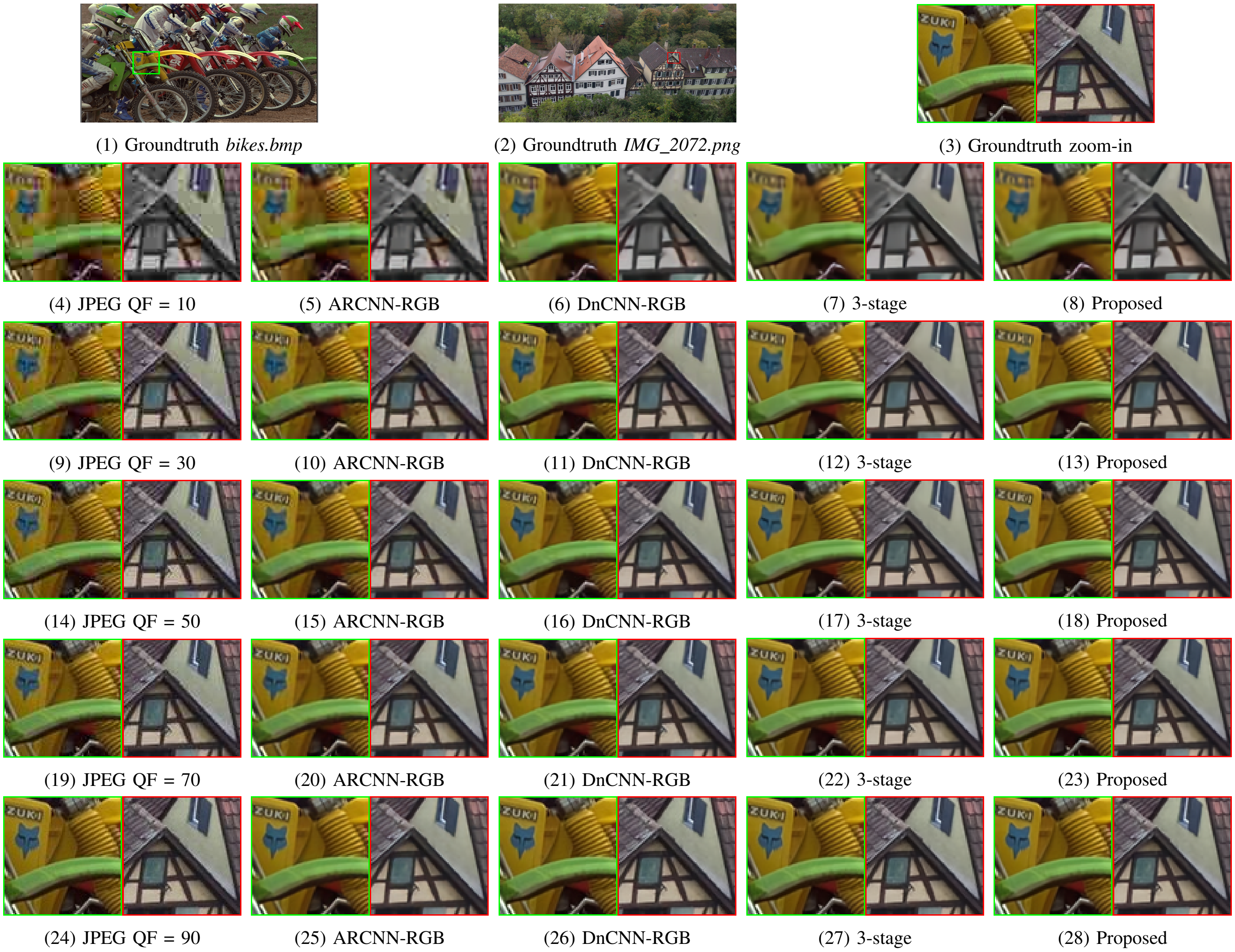
The average PSNR, MS-SSIM are adopted as objective evaluation metrics. Since blocking is one of the most important compression artifacts, average PSNR-B is also adopted, which is a specially designed metric to evaluate the block artifacts in a compressed image. $\mathrm{QF} \in [0..100]$ with a step size of 10 is used to evaluate the performance improvement for different quality levels. Tab. IV lists the results for these three metrics on the CLIC dataset.

As can be seen from Tab. IV, even a simple four-layer CNN, such as ARCNN-RGB covers a wide range of QF values and brings improvements. However, compared to the other approaches, the improvements are relatively small for all three metrics. DnCNN-RGB achieves better performance than ARCNN-RGB, because more layers are adopted. Our original 3-stage approach already outperforms the DnCNN-RGB by 0.1–0.2 dB regarding PSNR and achieves the best performance for PSNR-B. With the integration of the pixel-shuffle layer to our scheme, the proposed algorithm achieves an additional 0.1–0.2 dB PSNR improvement and corresponding MS-SSIM improvement. The PSNR-B performance of the proposed approach is comparable to the original 3-stage approach, which means that the pixel-shuffle structure maintains similar performance while significantly reducing the computational complexity. The proposed approach leads to the best performance for PSNR and MS-SSIM for all tested QF values.

In order to better visualize the improvements of different approaches for images compressed with different QF values, Fig. 10 presents the PSNR as a function of QF for the different approaches on the CLIC dataset. From Fig. 10, it can be seen that when the QF is very high, the PSNR of the enhanced images may even be lower than for the original JPEG images. For ARCNN-RGB, this happens when QF is greater than 85. It is mainly because of the limitation of ARCNN itself. It has a very limited number of trainable parameters. When the compressed image quality is high, it considers some high-frequency details as artifacts and smoothes them out. For DnCNN-RGB and the 3-stage approach, this happens only when the QF is very close to 100. The proposed approach brings consistent improvements for the whole QF range. Compared to the JPEG baseline, the PSNR is improved by 1.2–1.8 dB for a wide range of QF values.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148



(1) Groundtruth *bikes.bmp* (2) Groundtruth *IMG_2072.png* (3) Groundtruth zoom-in

(4) JPEG QF = 10 (5) ARCNN-RGB (6) DnCNN-RGB (7) 3-stage (8) Proposed

(9) JPEG QF = 30 (10) ARCNN-RGB (11) DnCNN-RGB (12) 3-stage (13) Proposed

(14) JPEG QF = 50 (15) ARCNN-RGB (16) DnCNN-RGB (17) 3-stage (18) Proposed

(19) JPEG QF = 70 (20) ARCNN-RGB (21) DnCNN-RGB (22) 3-stage (23) Proposed

(24) JPEG QF = 90 (25) ARCNN-RGB (26) DnCNN-RGB (27) 3-stage (28) Proposed

Fig. 9. Visual quality comparison on image *bikes.bmp* of the LIVE1 dataset and the *IMG_2072.png* of the CLIC dataset (best seen on a computer monitor).

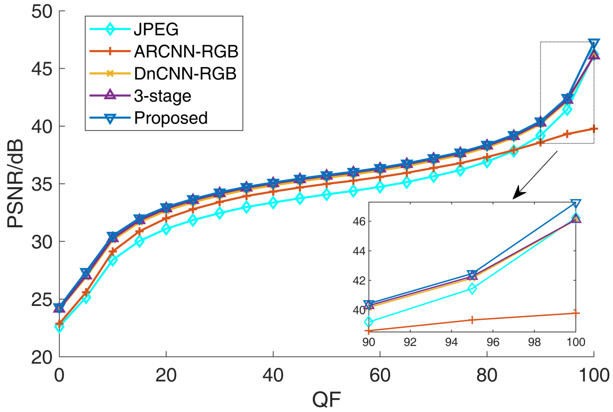


Fig. 10. PSNR performance as a function of QF on the CLIC dataset.

C. Real-world Color Image Denoising

For the RIDN task, the validation subset of the SIDD dataset is adopted to evaluate the performance of different algorithms. This dataset consists of 1280 noisy/clean image patch pairs of size 256 × 256 cropped from 40 high-resolution validation image pairs. We compare the proposed approach with several state-of-the-art approaches including TWSC [35], CDnCNN-B [6], CBDNet [38] and RIDNet [39]. The results here are generated using the source code downloaded from the original project pages. CDnCNN-B is one of the variants of DnCNN, which is targeting the blind color image denoising task for synthetic AWGN noise. In order to make a fair comparison, we retrain it with the SIDD dataset, which is noted as DnCNN-RGB in the results. Other parameters are the same as presented in the original paper.

Fig. 11 compares the visual quality of the different approaches. The noisy images suffer from strong noise. TWSC performs well for patch-wise smooth images. However, details are smoothed out when the image contains edges or textures. CDnCNN-B brings little improvements, as it is trained for synthetic AWGN noise. The retrained version DnCNN-RGB and CBDNet perform better, but obvious artifact patterns are observed. The RIDNet and the proposed approach generate clean output, and the proposed approach better reconstructs details.

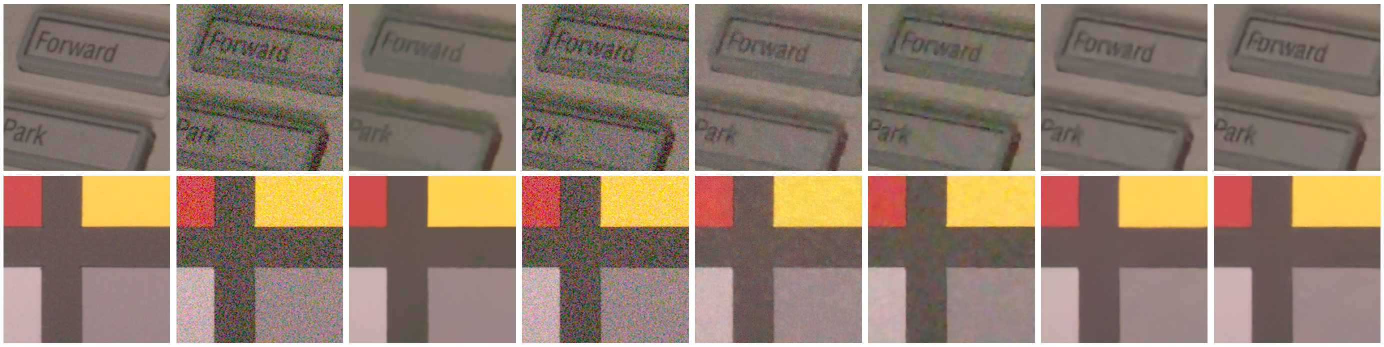
Tab. V presents the average PSNR and MS-SSIM values.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

TABLE IV
AVERAGE PSNR (IN DB), PSNR-B (IN DB) AND MS-SSIM RESULTS FOR THE JPEG CAR TASK (THE CLIC VALIDATION DATASET)

QF of JPEG	JPEG	ARCNN-RGB	DnCNN-RGB	3-stage	Proposed
0	22.61 / 25.74 / 0.7578	22.86 / 26.00 / 0.7681	24.11 / 27.33 / 0.8231	24.16 / 27.54 / 0.8279	24.31 / 27.52 / 0.8331
10	28.41 / 30.94 / 0.9036	29.14 / 31.90 / 0.9191	30.12 / 32.96 / 0.9416	30.27 / 33.21 / 0.9440	30.51 / 33.25 / 0.9475
20	31.10 / 33.37 / 0.9496	32.01 / 34.60 / 0.9597	32.67 / 35.27 / 0.9676	32.86 / 35.50 / 0.9689	32.99 / 35.51 / 0.9703
30	32.46 / 34.70 / 0.9654	33.42 / 36.04 / 0.9723	33.98 / 36.53 / 0.9765	34.17 / 36.75 / 0.9774	34.26 / 36.75 / 0.9783
40	33.38 / 35.61 / 0.9730	34.32 / 36.99 / 0.9783	34.85 / 37.39 / 0.9811	35.03 / 37.61 / 0.9818	35.11 / 37.59 / 0.9824
50	34.08 / 36.36 / 0.9778	34.99 / 37.74 / 0.9819	35.52 / 38.08 / 0.9841	35.69 / 38.30 / 0.9847	35.76 / 38.28 / 0.9851
60	34.74 / 37.09 / 0.9811	35.59 / 38.43 / 0.9844	36.14 / 38.76 / 0.9863	36.31 / 38.96 / 0.9868	36.38 / 38.94 / 0.9872
70	35.65 / 38.10 / 0.9850	36.37 / 39.33 / 0.9871	37.00 / 39.69 / 0.9889	37.16 / 39.87 / 0.9892	37.24 / 39.84 / 0.9895
80	36.92 / 39.57 / 0.9888	37.32 / 40.47 / 0.9898	38.15 / 40.99 / 0.9915	38.30 / 41.16 / 0.9917	38.40 / 41.09 / 0.9919
90	39.18 / 42.43 / 0.9930	38.59 / 41.99 / 0.9924	40.17 / 43.48 / 0.9944	40.29 / 43.60 / 0.9945	40.43 / 43.45 / 0.9946
100	46.23 / 53.95 / 0.9985	39.78 / 43.19 / 0.9944	46.10 / 52.74 / 0.9984	46.11 / 52.36 / 0.9983	47.27 / 52.84 / 0.9989



(a) Groundtruth (b) Noisy (c) TWSC (d) CDnCNN-B (e) CDnCNN-RT (f) CBDNet (g) RIDNet (h) Proposed

Fig. 11. Visual quality comparison on images patches of the SIDD validation dataset (best seen on a computer monitor).

TABLE V
AVERAGE PSNR, CPSNR (IN DB) AND MS-SSIM RESULTS FOR THE SIDD VALIDATION DATASET

Methods	R	G	B	RGB
Noisy	23.17	26.41	22.68	23.66 / 0.6237
TWSC [35]	35.12	37.49	35.04	35.53 / 0.9347
CDnCNN-B [6]	23.20	26.46	22.80	23.74 / 0.6242
DnCNN-RGB	30.91	32.57	30.77	31.17 / 0.8788
CBDNet [38]	32.65	37.26	31.79	33.07 / 0.9059
RIDNet [39]	38.31	40.84	38.05	38.71 / 0.9565
Proposed	38.73	40.99	38.35	39.04 / 0.9576

Except for CDnCNN-B, all other approaches bring significant PSNR and MS-SSIM improvements compared to the original noisy image. TWSC performs well. However, it is much slower than the other approaches because it solves an iterative optimization problem. The performance of DnCNN-RGB improves a lot after retraining with the SIDD dataset, but it is still not satisfactory. CBDNet performs slightly better than DnCNN. RIDNet outperforms the other approaches. Compared to RIDNet, the proposed approach brings additional PSNR and MS-SSIM improvements, especially for the red and blue channels.

VI. FURTHER DISCUSSION AND ANALYSIS

A. Computational Complexity Analysis

Computational complexity and running time are also critical for CIR tasks. The running time varies significantly with different hardware setups. To make a fair comparison, we take the CAR task as an example, in which the comparison approaches are re-implemented and re-trained for color images with the same hardware and software configurations. Since CNN-based approaches benefit significantly from GPU acceleration, we adopt both Intel i7-8700k Hexa-core CPU and Nvidia GeForce GTX 1080 Ti GPU for testing.

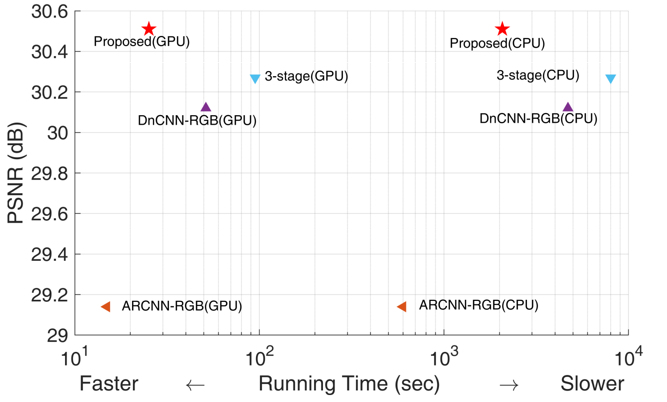


Fig. 12. PSNR performance as a function of the running time. The chart is based on the total running time in seconds for the CLIC dataset and PNSR values of QF = 10 in Tab. IV.

Fig. 12 shows the PSNR performance as a function of the running time of various approaches. We take the CLIC dataset and QF = 10 as an example. The closer to the upper-left corner, the better the algorithm performs. The CNN-based methods benefit a lot from GPU acceleration, usually more than 50 times faster than running on the CPU. ARCNN-

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

RGB is very fast, but it also has a significant performance drop. The running time of the original 3-stage approach is double compared to DnCNN. The proposed approach not only achieves state-of-the-art performance but also is three times faster compared to the original 3-stage approach. Even compared to DnCNN, the running time is only half.

B. Pixel-shuffle Convolution and Kernel Size

Another major advantage of the pixel-shuffle convolution is that it better explores the spatial and context information for the same kernel size and parameter settings. In order to prove this, different kernel sizes are adopted for the original 3-stage and the proposed improved 3-stage variants for the CDM task. These models are retrained with the same aforementioned parameter settings.

TABLE VI
AVERAGE PSNR AND CPSNR RESULTS (IN DB) OF VARIOUS KERNEL SIZES FOR THE WED-NEW DATASETS

Methods	R	G	B	RGB
3-stage (3×3)	39.32	43.04	39.37	40.19
3-stage (5×5)	39.79	43.41	39.75	40.61
Proposed (3×3)	40.04	43.60	39.98	40.84
Proposed (5×5)	40.11	43.65	40.05	40.90

As shown in Tab. VI, in general, larger kernel sizes bring better performance. For the original 3-stage approach, an additional 0.4dB PSNR improvement is observed. However, for the proposed approach, increasing the kernel size from 3×3 to 5×5, only a very small PSNR difference can be observed (less than 0.1dB), which means that most of the achievable improvement is already observed for the smaller kernel size. Also, the 3-stage pixel-shuffle CNN with 3×3 kernel achieves even better performance than the original 3-stage approach with 5×5 kernel, which proves that the pixel-shuffle CNN effectively and efficiently extracts spatial information.

C. The Effectiveness of the 3-stage Structure

Compared to a plain structure neural network, one of the key features of the proposed approach is the 3-stage CNN design. To prove that the reported improvements stem from the 3-stage structure rather than from the increasing number of parameters, we test a plain network structure that has a similar number of parameters without the 3-stage design. Without the 3-stage design, our model degrades to a DnCNN-like structure with the pixel-shuffle layer and three-channel color images as input and output.

In Tab. IV and V, we have shown that the proposed approach outperforms DnCNN for both CAR and RIDN tasks. There are 20 convolutional layers adopted in DnCNN, as suggested in the original paper for color image denoising tasks. To make the number of parameters roughly the same, we increase the number of convolutional layers in DnCNN to 35 and adopt the pixel-shuffle layer for the input and output images. This modified DnCNN is noted as *DnCNN-RGB+*. The modified network is retrained for RIDN and CAR tasks, and the results for each task are shown in Tab. VII and VIII, respectively.

TABLE VII
AVERAGE PSNR, CPSNR (IN DB) AND MS-SSIM RESULTS OF DNCNN-RGB+ FOR RIDN TASK (THE SIDD VALIDATION DATASET).

Methods	R	G	B	RGB
DnCNN-RGB (20 layers)	30.91	32.57	30.77	31.17 / 0.8788
DnCNN-RGB+ (35 layers)	35.31	35.52	35.40	35.31 / 0.9386
Proposed	38.73	40.99	38.35	39.04 / 0.9576

The results in Tab. VII show that the DnCNN-RGB+ brings additional performance improvement compared to the original DnCNN for the RIDN task. However, compared to TWSC, RIDNet, and the proposed method, DnCNN-RGB+ still performs worse. Considering the channel-wise reconstruction quality, the deeper model DnCNN-RGB+ brings the PSNR values for all three color channels to a similar level (about 35dB). For all other approaches, the reconstructed green channel always has better PSNR values. The possible reason is that DnCNN-RGB+ is a plain structure, and the three channels are reconstructed jointly. All the feature maps extracted are shared among all three channels and contribute to reconstructing these three channels. However, as analyzed in Tab. II, the color channels of real-world noisy images suffer from different distortion levels and require different compensation generated from the feature maps. One shared group of feature maps is not optimal to compensate for different levels of distortion.

TABLE VIII
AVERAGE PSNR (IN DB) RESULTS OF DNCNN-RGB+ FOR JPEG CAR TASK (THE LIVE1 DATASET).

QF of JPEG	DnCNN-RGB (20 layers)	DnCNN-RGB+ (35 layers)	Proposed
0	22.28	22.59	22.45
10	27.21	27.35	27.49
20	29.60	29.65	29.86
30	30.93	30.93	31.17
40	31.85	31.81	32.09
50	32.59	32.52	32.83
60	33.30	33.19	33.54
70	34.25	34.09	34.50
80	35.58	35.32	35.84
90	37.86	37.35	38.15
100	42.77	41.92	43.99

Tab. VIII presents the performance of the deeper model DnCNN-RGB+ compared to the original DnCNN and the proposed approach for the CAR task. For low quality levels ($\text{QF} = \{0, 10, 20\}$), DnCNN-RGB+ perform slightly better than DnCNN-RGB. When QF comes to 0, it performs better than the proposed approach. This is because, for extremely low quality, the images are highly distorted to a minimal number of colors or even gray-scale blocking images. Exploiting the inter-channel correlation in this case is less beneficial.

For medium to high quality levels ($\text{QF} \in [40..100]$), compared to DnCNN-RGB and the proposed network, DnCNN-RGB+ even suffers from some performance loss. On the one hand, a deeper model is usually more difficult to optimize and suffers from degradation issues. On the other hand, when quality levels become high, the inter-channel correlation becomes more critical to achieve accurate reconstruction. The proposed

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

3-stage network fully exploits the inter-channel correlation and benefits from it.

Another important characteristic of the proposed approach is that the green channel is reconstructed first, and then the red and blue channels are reconstructed with the guidance of the reconstructed green channel. To prove that the reconstruction of the green channel in the first stage is necessary, we reorder the reconstruction of color channels and perform an experiment with the CDM task.

In this experiment, the red channel is reconstructed in the first stage. In the second stage, the reconstructed red channel is adopted to guide the reconstruction of the green and blue channels in two parallel networks. All the intermediate reconstructed color components are concatenated and further refined in the third stage. This reordered version is noted as *Proposed-reorder*. Other settings and the number of trainable parameters remain the same. The experimental results are shown in Tab. IX.

TABLE IX
AVERAGE PSNR AND CPSNR RESULTS (IN DB) OF THE REORDERED 3-STAGE CNN FOR THE WED-NEW DATASETS

Methods	R	G	B	RGB
Proposed	40.04	43.60	39.98	40.84
Proposed-reorder	39.62	43.22	39.54	40.42

The results show that with the same network topology, reordering the reconstruction of the color channels results in more than 0.4dB PSNR loss. In the second stage of *Proposed-reorder*, the red channel is adopted to guide the reconstruction of the green and blue channels, which means the R/G and R/B correlation is considered and the G/B correlation is not well-exploited. From the analysis in Tab. I and Fig. 1, the correlation of G/B is much higher than that of R/B. Even this can be partiality compensated in the third stage when jointly refining, but there is still a noticeable performance loss. It proves that the reconstruction of the green channel in the first stage is necessary, and using two separate networks to explore the R/G, G/B correlation in the second stage is beneficial.

D. Analysis of the Quality-blind Training for CAR tasks

By introducing the quality-blind training strategy to the CAR task, a single trained model covers the whole quality range of JPEG and brings consistent quality improvement. It simplifies the training process significantly and makes the approach more practical in a real codec. In order to analyze the performance compromise of this strategy for different approaches, we have trained separate models for some QF values ($\mathrm{QF} \in [0..100]$ with a step size of 10). The comparison results with the LIVE1 dataset are shown in Tab. X. There are three columns of results under each approach. The column *blind* represents the performance with the quality-blind training strategy. The column *separate* indicates the performance of the model specially trained for the specific quality level. The column *gain* is the PSNR value difference between the separately training and the quality-blind training.

As shown in Tab. X, considering extreme low and high quality levels ($\mathrm{QF} = \{0, 90, 100\}$), there are noticeable PSNR drops when using quality-blind training strategy. ARCNN-RGB has the largest performance compromise compared to the others, and the proposed approach has the least performance loss.

Considering the quality range $\mathrm{QF} \in [10..80]$, for ARCNN-RGB, separate training brings additional 0.2–0.6dB PSNR improvements. The main reason is that the model capacity and trainable parameters in ARCNN-RGB are very limited (only four convolutional layers), which cannot cover so many different quality levels with a single trained model. However, compared to the JPEG baseline, the quality-blind trained ARCNN-RGB model still brings most of the achievable improvement. For DnCNN-RGB, the performance difference between separate training and quality-blind training is about 0.1dB. The proposed approach narrows the difference to less than 0.1dB for a wide range of quality levels. These results prove that this simple quality-blind training strategy is effective, and it leads to minimal performance compromise.

VII. CONCLUSION

This paper presents a 3-stage CNN-based color image restoration scheme exploiting the inter-channel correlation. With detailed theoretical and experimental analyses, we demonstrate that the inter-channel correlation has special characteristics, and the green channel has the best quality among all three channels in many distorted images. Based on these characteristics, a 3-stage CNN scheme is proposed, in which the inter-channel correlation is fully exploited stage by stage. Stage one is designed to reconstruct the green channel. In the second stage, the red and blue channels are enhanced with two parallel networks with the guidance of the high-quality green channel obtained from the first stage. The intermediate results are concatenated and further refined in the third stage. By integrating the pixel-shuffle convolution to the 3-stage network, both the efficiency and the performance of the network are significantly improved. By introducing the quality-blind training strategy to the CAR task, the whole quality range of JPEG is covered by a single trained model, which makes the training more manageable and the approach more practical. Extensive experiments on different CIR tasks show that the proposed scheme can be applied for various CIR tasks and outperforms the state-of-the-art approaches.

REFERENCES

[1] L. Zhang and W. Zuo, “Image restoration: From sparse and low-rank priors to deep priors [lecture notes],” *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 172–179, Sept. 2017.

[2] W. Dong, G. Shi, and X. Li, “Nonlocal image restoration with bilateral variance estimation: A low-rank approach,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 700–711, Feb. 2013.

[3] J. Zhang, D. Zhao, and W. Gao, “Group-based sparse representation for image restoration,” *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.

[4] N. Eslahi and A. Aghagolzadeh, “Compressive sensing image restoration using adaptive curvelet thresholding and nonlocal sparse regularization,” *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3126–3140, July 2016.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

TABLE X
AVERAGE PSNR (IN DB) RESULTS OF QUALITY-BLIND TRAINING AND SEPARATE TRAINING FOR EACH QF (THE LIVE1 DATASET). SEPARATE TRAINING FOR EACH QF RESULTS ARE MARKED IN CYAN AND THE PSNR GAINS BETWEEN SEPARATE AND QUALITY-BLIND TRAINING ARE MARKED IN MAGENTA

QF of JPEG	JPEG	ARCNN-RGB			DnCNN-RGB			Proposed		
		blind	separate	gain	blind	separate	gain	blind	separate	gain
0	20.89	21.16	22.38	1.22	22.28	22.62	0.34	22.45	22.65	0.20
10	25.69	26.37	26.93	0.56	27.21	27.29	0.08	27.49	27.51	0.02
20	28.06	28.91	29.28	0.37	29.60	29.68	0.08	29.86	29.89	0.03
30	29.37	30.26	30.53	0.27	30.93	31.01	0.08	31.17	31.21	0.04
40	30.28	31.18	31.43	0.25	31.85	31.94	0.09	32.09	32.13	0.04
50	31.03	31.90	32.12	0.22	32.59	32.68	0.09	32.83	32.89	0.06
60	31.77	32.58	32.81	0.23	33.30	33.42	0.12	33.54	33.63	0.09
70	32.77	33.44	33.70	0.36	34.25	34.39	0.14	34.50	34.58	0.08
80	34.23	34.52	35.02	0.50	35.58	35.77	0.19	35.84	35.96	0.12
90	36.86	35.96	37.40	1.44	37.86	38.38	0.52	38.15	38.47	0.32
100	43.07	37.13	40.59	3.46	42.77	44.57	1.80	43.99	44.69	0.70

[6] 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, July 2017.

[7] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, Sept. 2018.

[8] Z. Jin, M. Z. Iqbal, D. Bobkov, W. Zou, X. Li, and E. Steinbach, "A flexible deep CNN framework for image restoration," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 1055–1068, Apr. 2020.

[9] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2305–2318, Oct. 2019.

[10] B. Zheng, Y. Chen, X. Tian, F. Zhou, and X. Liu, "Implicit dual-domain convolutional network for robust color image compression artifact reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2019.

[11] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space," in *2007 IEEE International Conference on Image Processing*, vol. 1, Sept. 2007, pp. 313–316.

[12] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[13] Y. Wen, M. K. Ng, and Y. Huang, "Efficient total variation minimization methods for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2081–2088, Nov. 2008.

[14] M. Jung, X. Bresson, T. F. Chan, and L. A. Vese, "Nonlocal mumford-shah regularizers for color image restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1583–1598, June 2011.

[15] K. Cui, Z. Jin, and E. Steinbach, "Color image demosaicking using a 3-stage convolutional neural network structure," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 2177–2181.

[16] K. Cui and E. Steinbach, "Decoder side image quality enhancement exploiting inter-channel correlation in a 3-stage CNN: Submission to CLIC 2018," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018, pp. 2571–2574.

[17] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1874–1883.

[18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, vol. 6064, International Society for Optics and Photonics. SPIE, Feb. 2006, pp. 354 – 365.

[19] A. Rajwade, A. Rangarajan, and A. Banerjee, "Image denoising using the higher order singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 849–862, Apr. 2013.

[20] H. S. Mousavi and V. Monga, "Sparsity-based color image super resolution via exploiting cross channel constraints," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5094–5106, Nov. 2017.

[21] T. Miyata, "Inter-channel relation based vectorial total variation for color image recovery," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sept. 2015, pp. 2251–2255.

[22] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2808–2817.

[23] B. E. Bayer, "Color imaging array," Jul. 20 1976, US Patent 3,971,065.

[24] D. Menon and G. Calvagno, "Color image demosaicking: An overview," *Signal Processing: Image Communication*, vol. 26, no. 8, pp. 518–533, Oct. 2011.

[25] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 191:1–191:12, Nov. 2016.

[26] F. Kokkinos and S. Lefkimmiatis, "Iterative joint image demosaicking and denoising using a residual denoising network," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4177–4188, Aug. 2019.

[27] R. Tan, K. Zhang, W. Zuo, and L. Zhang, "Color image demosaicking via deep residual learning," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 793–798.

[28] D. S. Tan, W. Chen, and K. Hua, "Deepdemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2408–2419, May 2018.

[29] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 576–584.

[30] K. Cui and E. Steinbach, "Decoder side color image quality enhancement using a wavelet transform based 3-stage convolutional neural network," in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[31] H. Chen, X. He, L. Qing, S. Xiong, and T. Q. Nguyen, "DPW-SDNet: Dual pixel-wavelet domain deep cnns for soft decoding of JPEG-compressed images," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018, pp. 824–833.

[32] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-CNN for image restoration," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018, pp. 886–894.

[33] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1683–1691.

[34] J. Xu, L. Zhang, D. Zhang, and X. Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 1105–1113.

[35] J. Xu, L. Zhang, and D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer International Publishing, Sept. 2018, pp. 21–38.

[36] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 1692–1700.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/JSTSP.2020.3043148

[37] T. Plötz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2750–2759.

[38] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 1712–1722.

[39] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *2019 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 3155–3164.

[40] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk, "What is the space of spectral sensitivity functions for digital color cameras?" in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, Jan. 2013, pp. 168–179.

[41] J. Y. Hardeberg, H. Brettel, and F. J. M. Schmitt, "Spectral characterization of electronic cameras," in *Electronic Imaging: Processing, Printing, and Publishing in Color*, vol. 3409, International Society for Optics and Photonics. SPIE, Sept. 1998, pp. 100 – 109.

[42] S. Han, Y. Matsushita, I. Sato, T. Okabe, and Y. Sato, "Camera spectral sensitivity estimation from a single image under unknown illumination by using fluorescence," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2012, pp. 805–812.

[43] "Kodak lossless true color image suite," http://http://r0k.us/graphics/kodak/.

[44] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic Imaging*, vol. 20, no. 2, pp. 1 – 17 – 17, Apr. 2011.

[45] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017, pp. 1122–1131.

[46] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, Feb. 2017.

[47] D. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proceedings of the 6th ACM Multimedia Systems Conference*, Mar. 2015, p. 219–224.

[48] ITU-T T.871, "Information technology – digital compression and coding of continuous-tone still images: JPEG file interchange format (JFIF)," May, 2011.

[49] W. Luo, J. Huang, and G. Qiu, "Jpeg error analysis and its applications to digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 480–491, Sept. 2010.

[50] ITU-R BT.601-7, "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios," Mar., 2011.

[51] ITU-R BT.709-6, "Parameter values for the HDTV standards for production and international programme exchange," June, 2015.

[52] ITU-R BT.2020-2, "Parameter values for ultra-high definition television systems for production and international programme exchange," Oct., 2015.

[53] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik, "Live image quality assessment database release 2," http://live.ece.utexas.edu/research/quality.

[54] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, Nov. 2003, pp. 1398–1402.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1026–1034.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *2015 International Conference on Learning Representations (ICLR)*, May 2015.

[57] R. Timofte, R. Rothe, and L. V. Gool, "Seven ways to improve example-based single image super resolution," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1865–1873.

[58] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, Mar. 2005.

[59] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2167–2178, Dec. 2005.

[60] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-laplacian residual interpolation for color image demosaicking," in *Proceedings of SPIE 9023, Digital Photography X*, vol. 9023, Mar 2014, p. 90230L.

[61] ——, "Beyond color difference: Residual interpolation for color image demosaicking," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1288–1300, Mar. 2016.

[62] Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, "Adaptive residual interpolation for color and multispectral image demosaicking," *Sensors*, vol. 17, no. 12, Dec. 2017.

[63] CLIC, "Workshop and challenge on learned image compression," https://www.compression.cc/.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.