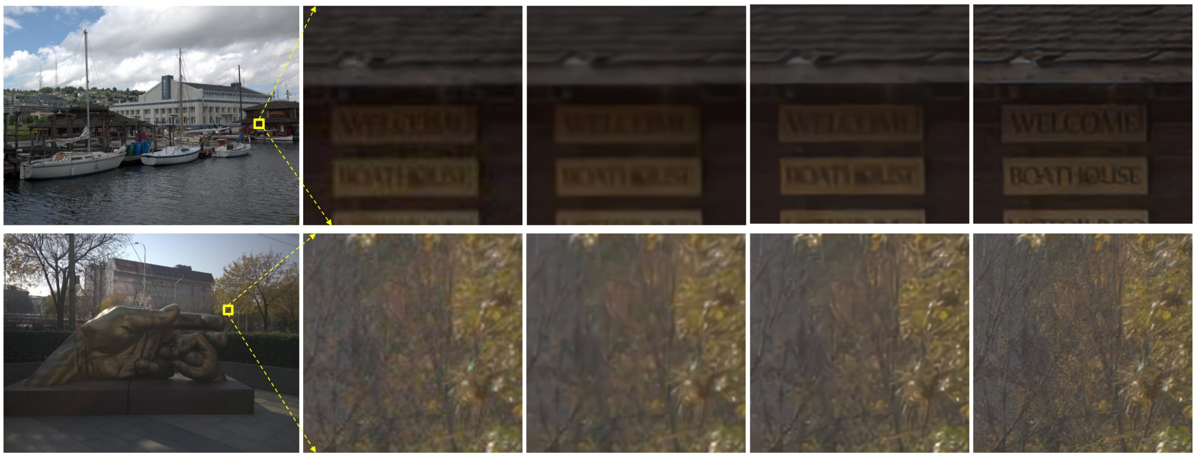
Towards Real Scene Super-Resolution with Raw Images

Xiangyu Xu Yongrui Ma Wenxiu Sun
SenseTime Research



(a) Whole input image (b) Input (c) RDN with previous data (d) RDN with our data (e) Ours

Figure 1. Existing super-resolution method (RDN [38]) does not perform well for real captured images as shown in (c). We can obtain sharper results (d) by re-training existing model [38] with the data generated by our method. Furthermore, we recover more structures and details (e) by exploiting the radiance information recorded in raw images. The two input images (a) are captured by Leica SL Typ-601 and iPhone 6s respectively, and both cameras are not seen by the models during training. Results best viewed electronically with zoom.

Abstract

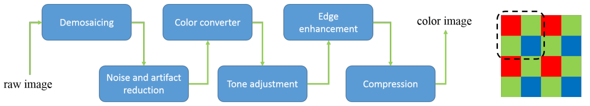
Most existing super-resolution methods do not perform well in real scenarios due to lack of realistic training data and information loss of the model input. To solve the first problem, we propose a new pipeline to generate realistic training data by simulating the imaging process of digital cameras. And to remedy the information loss of the input, we develop a dual convolutional neural network to exploit the originally captured radiance information in raw images. In addition, we propose to learn a spatially-variant color transformation which helps more effective color corrections. Extensive experiments demonstrate that super-resolution with raw data helps recover fine details and clear structures, and more importantly, the proposed network and data generation pipeline achieve superior results for single image super-resolution in real scenarios.

1. Introduction

In optical photography, the number of pixels representing an object, *i.e*. the image resolution, is directly proportional to square of the focal length of the camera [11]. While one can use a long-focus lens to obtain a high-resolution image, the range of the captured scene is usually limited by the size of the sensor array at the image plane. Thus, it is often desirable for users to capture the wide-range scene at a lower resolution with a short-focus camera (*e.g*. a wide-angle lens), and then apply the single image super-resolution technique which recovers a high-resolution image from its low-resolution version.

Most state-of-the-art super-resolution methods [4, 18, 31, 34, 20, 32, 38, 37] are based on data-driven models, and in particular, deep convolutional neural networks (CNNs) [4, 27, 21]. While these methods are effective on synthetic data, they do not perform well for real captured images by cameras or cellphones (examples shown in Figure 1(c)) owing to lack of realistic training data and information loss of network input. To remedy these problems and enable real scene super-resolution, we propose a new pipeline for generating training data and a dual CNN model for exploiting additional raw information, which are described as below.

First, most existing methods cannot synthesize realistic training data; the low-resolution images are usually generated with a fixed downsampling blur kernel (*e.g*. bicubic kernel) and homoscedastic Gaussian noise [38, 35]. On one hand, the blur kernel in practice may vary with zoom, fo-



(a) ISP pipeline (b) Bayer pattern

Figure 2. Typical ISP pipeline of digital cameras. (b) represents a Bayer pattern raw image, where the sensels surrounded by the black-dot curve is a Bayer pattern block.

cus, and camera shake during image capturing, which is beyond the fixed kernel assumption. On the other hand, image noise usually obeys heteroscedastic Gaussian distribution [26] whose variance depends on the pixel intensity, which is in sharp contrast to the homoscedastic Gaussian noise. More importantly, both the blur kernel and noise should be applied to the linear raw data whereas previous approaches use the pre-processed non-linear color images. To solve above problems, we synthesize the training data in linear space by simulating the imaging process of digital cameras, and applying different kernels and heteroscedastic Gaussian noise to approximate real scenarios. As shown in Figure 1(d), we can obtain sharper results by training existing model [38] with the data from our generation pipeline.

Second, while modern cameras provide both the raw data and the pre-processed color image (produced by the image signal processing system, *i.e.* ISP) to users [12], most super-resolution algorithms only take the color image as input, which does not make full use of the radiance information existing in raw data. By contrast, we directly use raw data for restoring high-resolution clear images, which conveys several advantages: (i) More information could be exploited in raw pixels since they are typically 12 or 14 bits [1], whereas the color pixels produced by ISP are typically 8 bits [12]. We show a typical ISP pipeline in Figure 2(a). Except for the bit depth, there is additional information loss within the ISP pipeline, such as the noise reduction and compression [25]. (ii) Raw data is proportional to scene radiance, while the ISP contains nonlinear operations, such as tone mapping. Thus, the linear degradations in the imaging process, including blur and noise, are nonlinear in the processed RGB space, which brings more difficulties in image restoration [24]. (iii) The demosaicing step in the ISP is highly related to super-resolution, because these two problems both refer to the resolution limitations of cameras [39]. Therefore, to solve the super-resolution problem with pre-processed images is sub-optimal and could be inferior to a single unified model solving both problems simultaneously.

In this paper, we introduce a new super-resolution method to exploit raw data from camera sensors. Existing raw image processing networks [2, 29] usually learn a direct mapping function from the degraded raw image to the desired full color output. However, raw data does not have the relevant information for color corrections conducted within the ISP system, and thereby the networks trained with it could only be used for one specific camera. To solve this problem, we propose a dual CNN architecture (Figure 3) which takes both the degraded raw and color images as input, so that our model could generalize well to different cameras. The proposed model consists of two parallel branches, where the first branch restores clear structures and fine details with the raw data, and the second branch recovers high-fidelity colors with the low-resolution RGB image as reference. To exploit multi-scale features, we use densely connected convolution layers [15] in an encoder-decoder framework for image restoration. For the color correction branch, simply adopting the technique in [29] to learn a global transformation usually leads to artifacts and incorrect color appearances. To address this issue, we propose to learn pixel-wise color transformations to handle more complex spatially-variant color operations and generate more appealing results. In addition, we introduce feature fusion for more accurate color correction estimation. As shown in Figure 1(e), the proposed algorithm significantly improves the super-resolution results for real captured images.

The main contributions of this work are summarized as follows. First, we design a new data generation pipeline which enables synthesizing realistic raw and color training data for image super-resolution. Second, we develop a dual network architecture to exploit both raw data and color images for real scene super-resolution, which is able to generalize to different cameras. In addition, we propose to learn spatial-variant color transformations as well as feature fusion for better performance. Extensive experiments demonstrate that solving the problem using raw data helps recover fine details and clear structures, and more importantly, the proposed network and data generation pipeline achieve superior results for single image super-resolution in real scenarios.

2. Related Work

We discuss state-of-the-art super-resolution methods as well as learning-based raw image processing, and put this work in proper context.

Super-resolution. Most state-of-the-art super-resolution methods [34, 18, 32, 4, 38, 20, 6, 37] learn CNNs to reconstruct high-resolution images from low-resolution color inputs. Dong *et al.* [4] propose a three-layer CNN for mapping the low-resolution patches to high-resolution space, but fail to get better results with deeper networks [5]. To solve this problem, Kim *et al.* [18] introduce residual learning to accelerate training and achieve better results. Tong *et al.* [32] use dense skip connections to further speed up the reconstruction process. While these methods are effective in interpolating pixels, they are based on preprocessed color images and thus have limitations in producing realistic

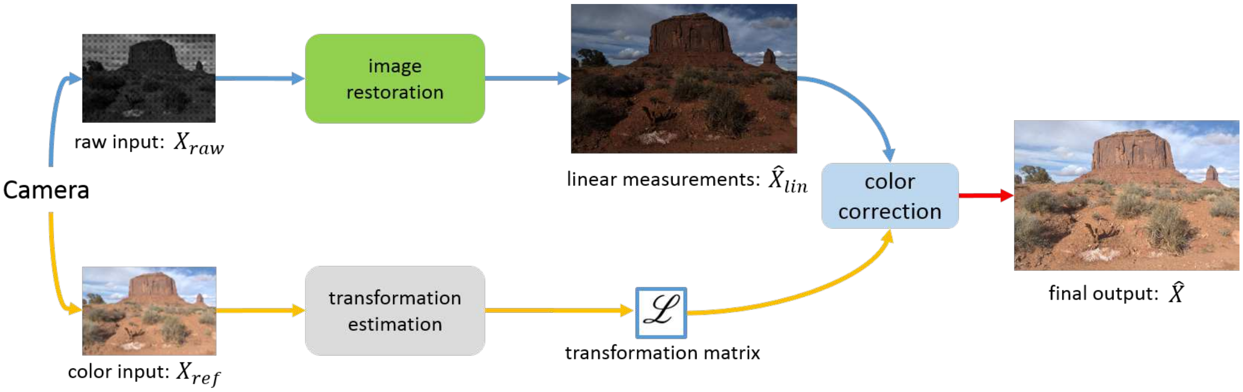


Figure 3. Overview of the proposed network. Our model has two parallel branches, where the first branch exploits raw data X_{raw} to restore high-resolution linear measurements $\widehat{X}_{lin}$ for all color channels with clear structures and fine details, and the second branch estimates the transformation matrix to recover the final color result $\widehat{X}$ using the low-resolution color image X_{ref} as reference.

details. By contrast, we propose to exploit both raw data and color image in a unified framework for better super-resolution.

Joint super-resolution and demosaicing. Many existing methods for this problem estimate a high-resolution color image with multiple low-resolution frames [7, 33]. More closely related to our task, Zhou *et al*. [39] propose a deep residual network for single image super-resolution with mosaiced images. However, this model is trained on gamma-corrected image pairs which may not work well for real linear data. More importantly, these works do not consider the complex color correction steps applied by camera ISPs, and thus cannot recover high-fidelity color appearances. Different from them, the proposed algorithm solves the problems of image restoration and color correction simultaneously, which are more suitable for real applications.

Learning-based raw image processing. In recent years, learning-based methods have been proposed for raw image processing [16, 2, 29]. Jiang *et al*. [16] propose to learn a large collection of local linear filters to approximate the complex nonlinear ISP pipelines. Following their work, Schwartz *et al*. [29] use deep CNNs for learning the color correction operations of specific digital cameras. Chen *et al*. [2] train a neural network with raw data as input for fast low-light imaging. In this work, we learn color correction in the context of raw image super-resolution. Instead of learning a color correction pipeline for one specific camera, we use a low-resolution color image as reference for handling images from more diverse ISP systems.

3. Method

For better super-resolution results in real scenarios, we propose a new data generation pipeline to synthesize more realistic training data, and a dual CNN model to exploit the radiance information recorded in raw data. We describe the synthetic data generation pipeline in Section 3.1 and the network architecture in Section 3.2.

3.1. Training Data

Most super-resolution methods [18, 4] generate training data by downsampling high-resolution color images with a fixed bicubic blur kernel. And homoscedastic Gaussian noise is often used to model real image noise [38]. However, as introduced in Section 1, the low-resolution images generated in this way does not resemble real captured images and will be less effective for training real scene super-resolution models. Moreover, we need to generate low-resolution raw data as well for training the proposed dual CNN, which is often approached by directly mosaicing the low-resolution color images [9, 39]. This strategy ignores the fact that the color images have been processed by nonlinear operations of the ISP system while the raw data should be from linear color measurements of the pixels. To solve these problems, we start with high-quality raw images [1] and generate realistic training data by simulating the imaging process of the degraded images. We first synthesize ground truth linear color measurements by downsampling the high-quality raw images so that each pixel could have its ground truth red, green and blue values. In particular, for Bayer pattern raw data (Figure 2(b)), we define a block of Bayer pattern sensels as one new virtual sensel, where all color measurements are available. In this way, we can obtain the desired images $X_{lin} \in \mathbb{R}^{H\times W\times 3}$ with linear measurements of all three colors for each pixel. H and W denote the height and width of the ground truth linear image. Similar with [17], we compensate the color shift artifact by aligning the center of mass of each color in the new sensel. With the ground truth linear measurements X_{lin}, we can easily generate the ground truth color images

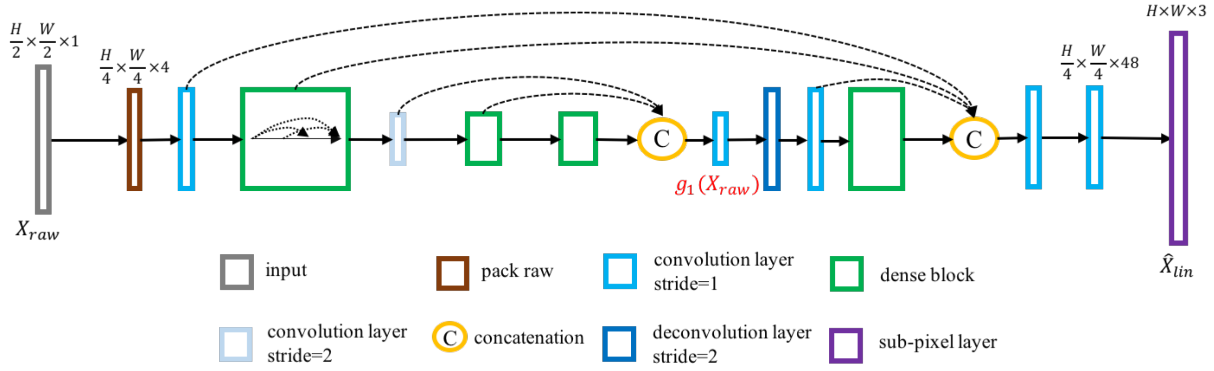

$\frac{H}{2}\times\frac{W}{2}\times 1$
$\frac{H}{4}\times\frac{W}{4}\times 4$
$H\times W\times 3$
$\frac{H}{4}\times\frac{W}{4}\times 48$
X_{raw}
C
C
$g_1(X_{raw})$
$\hat{X}_{lin}$
input
pack raw
convolution layer stride=1
dense block
convolution layer stride=2
concatenation
deconvolution layer stride=2
sub-pixel layer

Figure 4. The image restoration branch adopts dense blocks in an encoder-decoder framework, which reconstructs high-resolution linear color measurements $\widehat{X}_{lin}$ from the degraded low-resolution raw input X_{raw}.

$X_{gt} \in \mathbb{R}^{H\times W\times 3}$ by simulating the color correction steps of the camera ISP system, such as color space conversion and tone adjustment. For the simulation, we use Dcraw [3] which is a widely-used raw processing software.

To generate degraded low-resolution raw images $X_{raw} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}}$, we separately apply blurring, downsampling, Bayer sampling, and noise onto the linear color measurements:

$$X_{raw} = f_{Bayer}(f_{down}(X_{lin} * k_{def} * k_{mot})) + n, \quad (1)$$

where f_{Bayer} is the Bayer sampling function which mosaics images in accordance with the Bayer pattern (Figure 2(b)). f_{down} represents the downsampling function with a sampling factor of two. Since the imaging process is likely to be affected by out-of-focus effect as well as camera shake, we consider both defocus blur k_{def} modeled as disk kernels with variant sizes, and modest motion blur k_{mot} generated by random walk [28]. $*$ denotes the convolution operator. To synthesize more realistic training data, we add heteroscedastic Gaussian noise [23, 26, 13] to the generated raw data:

$$n(x) \sim \mathcal{N}(0, \sigma_1^2 x + \sigma_2^2), \quad (2)$$

where the variance of noise n depends on the pixel intensity x. σ_1 and σ_2 are parameters of the noise. Finally, the raw image X_{raw} is demosaiced with the AHD [14], a commonly-used demosaicing method, and further processed by Dcraw to produce the low-resolution color image $X_{ref} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$. We compress X_{ref} as 8-bit JPEG as normally done in digital cameras. Note that the settings for the color correction steps in Dcraw are the same as those used in generating X_{gt}, so that the reference colors correspond well to the ground truth. The proposed pipeline synthesizes realistic data so that the models trained on X_{raw} and X_{ref} generalize well to real degraded raw and color images (Figure 1(e)).

3.2. Network Architecture

A straightforward way to exploit raw data for super-resolution is to directly learn a mapping function from raw images to high-resolution color images with neural networks. While the raw data is advantageous for image restoration, this naive strategy does not work well in practice; because the raw data does not have the relevant information for color correction and tone enhancement which have been conducted within the ISP system of digital cameras. In fact, one raw image could potentially correspond to multiple ground truth color images generated by different image processing algorithms of various ISP pipelines, which will confuse the network training and make it non-practical to train a model to reconstruct high-resolution images with desired colors. To solve this problem, we propose a two-branch CNN as shown in Figure 3, where the first branch exploits raw data to restore high-resolution linear measurements for all color channels with clear structures and fine details, and the second branch estimates the transformation matrix to recover high-fidelity color appearances using the low-resolution color image as reference. The reference image complements the limitations of raw data; thus, jointly training two branches with raw and color data could essentially help recover better results.

Image restoration. We show an illustration of the image restoration network in Figure 4. Similar with [2], we first pack the raw data X_{raw} into four channels which respectively correspond to the R, G, B, G patterns in Figure 2(b). Then we apply a convolution layer to the packed four-channel input for learning low-level features, and consecutively adopt four dense blocks [15, 32] to learn more complex nonlinear mappings for image restoration. The dense block is composed of eight densely connected convolution layers. Different from previous work [15, 32], we deploy the dense blocks in an encoder-decoder framework to make the model more compact and efficient. We use

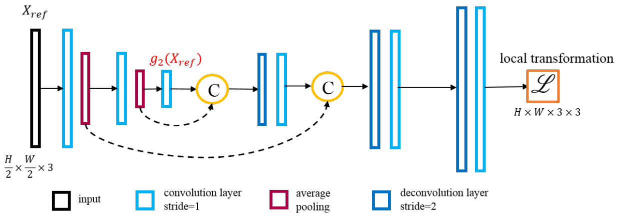


Figure 5. Network architecture of the color correction branch. Our model predicts the pixel-wise transformation $\mathscr{L}(X_{ref})$ with a reference color image.

$g_1(X_{raw})$ to denote the encoded features as shown in Figure 4. Finally, the feature maps with the same spatial size are concatenated and fed into a convolution layer, and this layer produces 48 feature maps which are rearranged by a sub-pixel layer [30] to restore the three-channel linear measurements $\widehat{X}_{lin} \in \mathbb{R}^{H \times W \times 3}$ with 2 times resolution of the input.

Color correction. With the predicted linear measurements $\widehat{X}_{lin}$, we learn the second branch for color correction which reconstructs the final result $\widehat{X} \in \mathbb{R}^{H \times W \times 3}$ using the color reference image X_{ref}. Similar with [29], we use a CNN to estimate a transformation $\mathscr{G}(X_{ref}) \in \mathbb{R}^{3 \times 3}$ to produce a global color correction of the image. Mathematically, the correction could be formulated as: $\mathscr{G}(X_{ref})\widehat{X}_{lin}[i,j]$, where $\widehat{X}_{lin}[i,j] \in \mathbb{R}^{3 \times 1}$ represents the RGB values at pixel $[i,j]$ of the linear image. However, this global correction strategy does not work well when the color transformation of camera ISP involves spatially-variant operations.

To solve this problem, we propose to learn a pixel-wise transformation $\mathscr{L}(X_{ref}) \in \mathbb{R}^{H \times W \times 3 \times 3}$ which allows different color corrections for each spatial location. Thus, we can generate the final results as:

$$\widehat{X}[i,j] = \mathscr{L}(X_{ref})[i,j]\widehat{X}_{lin}[i,j], \tag{3}$$

where $\mathscr{L}(X_{ref})[i,j] \in \mathbb{R}^{3 \times 3}$ is the local transformation matrix for the RGB vector at pixel $[i,j]$. Note that we directly transform the RGB vectors instead of using the quadratic form in [29] as we empirically find no benefits from it.

We adopt a U-Net structure [8, 36] for predicting the transformation $\mathscr{L}(X_{ref})$ in Figure 5. The CNN starts with an encoder including three convolution layers with 2 × 2 average pooling to extract the encoded features $g_2(X_{ref})$. To estimate the spatially-variant transformations, we adopt a decoder consisting of consecutive deconvolution and convolution layers, which expands the encoded features to the desired resolution $H \times W$. We concatenate the feature maps with the same resolution in the encoder and decoder to exploit hierarchy features. Finally, the output layer of the decoder generates 3 × 3 weights for each pixel, which form the pixel-wise transformation matrix $\mathscr{L}(X_{ref})$.

Feature fusion. As the transformation $\mathscr{L}(X_{ref})$ is applied to the restored image, it would be beneficial to make the matrix estimation network aware of the features in the image restoration branch, so that $\mathscr{L}(X_{ref})$ could more accurately adapt to the structures of the restored image. Towards this end, we develop feature fusion for $g_2(X_{ref})$ using the encoded features from the first branch $g_1(X_{raw})$. As $g_2(X_{ref})$ and $g_1(X_{raw})$ are likely to have different scales, we fuse them by weighted sum where the weights are learned to adaptively update the features. We could formulate the feature fusion as:

$$g_2(X_{ref}) \leftarrow g_2(X_{ref}) + \phi(g_1(X_{raw})), \tag{4}$$

where ϕ is a 1×1 convolution. After this, we put the updated $g_2(X_{ref})$ back into the original branch, and the following decoder could further use the fused features for transformation estimation. The convolutions are initialized with zeros to avoid interrupting the initial behavior of the color correction branch.

4. Experimental Results

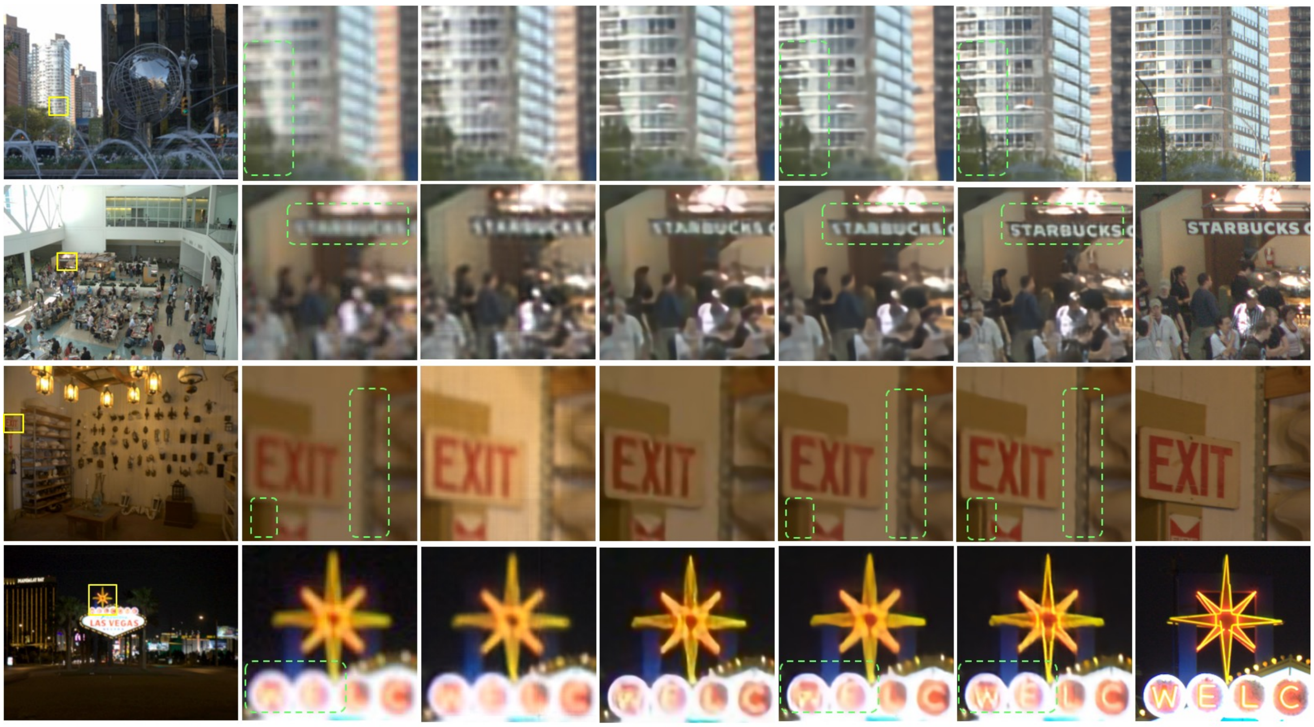
We describe the implementation details of our method and present analysis and evaluations on both synthetic and real data in this section.

4.1. Implementation Details

For generating training data, we use the MIT-Adobe 5K dataset [1] which is composed of 5000 raw photographs with size around 2000 × 3000. After manually removing images with noticeable noise and blur, we obtain 1300 high-quality images for training and 150 for testing which are captured by 19 types of Canon cameras. We use the method described in Section 3.1 to synthesize the training and test datasets. The radius of the defocus blur is randomly sampled from $[1, 5]$ and the maximum size of the motion kernel is sampled from $[3, 11]$. The noise parameters σ_1, σ_2 are respectively sampled from $[0, 10^{-2}]$ and $[0, 10^{-3}]$.

We use the Xavier initializer [10] for the network weights and the LeakyReLU [22] with slope 0.2 as the activation function. We adopt the L_1 loss function for training the network and use the Adam optimizer [19] with initial learning rate 2×10^{-4}. We crop 256 × 256 patches as input and use a batch size of 6. We first train the model with 4×10^4 iterations where the learning rate is decreased by a factor of 0.96 every 2×10^3 updates, and then another 4×10^4 iterations at a lower rate 10^{-5}. During the test phase, we chop the input into overlapping patches and process each patch separately. The reconstructed high-resolution patches are placed back to the corresponding locations and averaged in overlapping regions.

Baseline methods. We compare our method with state-of-the-art super-resolution methods [4, 18, 32, 38] as well as



(a) Whole input image (b) Input (c) SID (d) SRDenseNet (e) RDN (f) Ours (g) GT

Figure 6. Results on our synthetic dataset. References for the baseline methods could be found in Table 1. “GT” represent ground truth.

Table 1. Quantitative evaluations on the proposed synthetic dataset. “Blind” represents the images with variable blur kernels, and “Non-blind” denotes fixed kernel. Red and blue text indicates the first and second best performance, respectively.

Methods	Blind		Non-blind	
	PSNR	SSIM	PSNR	SSIM
SID [2]	21.87	0.7325	21.89	0.7346
DeepISP [29]	21.71	0.7323	21.84	0.7382
SRCNN [4]	28.83	0.7556	29.19	0.7625
VDSR [18]	29.32	0.7686	29.88	0.7803
SRDenseNet [32]	29.46	0.7732	30.05	0.7844
RDN [38]	29.93	0.7804	30.46	0.7897
w/o color branch	20.54	0.7252	21.13	0.7413
w/o local color correction	29.81	0.7954	30.29	0.8095
w/o raw input	30.05	0.7827	30.51	0.7921
w/o feature fusion	30.73	0.8025	31.68	0.8252
Ours full model	30.79	0.8044	31.79	0.8272

learning-based raw image processing algorithms [2, 29]. As the raw image processing networks [2, 29] are not originally designed for super-resolution, we add deconvolution layers after them for increasing the output resolution. All the baseline models are trained on our data with the same settings.

4.2. Results on Synthetic Data

We quantitatively evaluate our method using the test dataset described above. Table 1 shows that the proposed algorithm performs favorably against the baseline methods in terms of both PSNR and structural similarity (SSIM). Figures 6 shows some restored results from our synthetic dataset. Since the low-resolution raw input does not have the relevant information for color conversion and tone adjustment conducted within the ISP system, the raw image processing methods [2, 29] can only approximate the color corrections of one specific camera, and thereby cannot recover good results for different camera types in our test set. In addition, the training process of the raw processing models is influenced by the color differences between the predictions and ground truth, and cannot focus on restoring sharp structures. Thus, the results in Figures 6(c) are still blurry. Figure 6(d)-(e) show that the super-resolution methods with low-resolution color image as input generate shaper results with correct colors, which, however, still lack fine details. In contrast, our method achieves better results with clear structures and fine details in Figure 6(f) by exploiting both the color input and raw radiance information.

Non-blind super-resolution. Since most super-resolution methods [4, 18, 38] are non-blind with the assumption that the downsampling blur kernel is known and fixed, we also evaluate our method for this case by using fixed defocus kernel with radius 5 to synthesize training and test datasets. As shown in Table 1, the proposed method achieves competitive super-resolution results under non-blind settings.

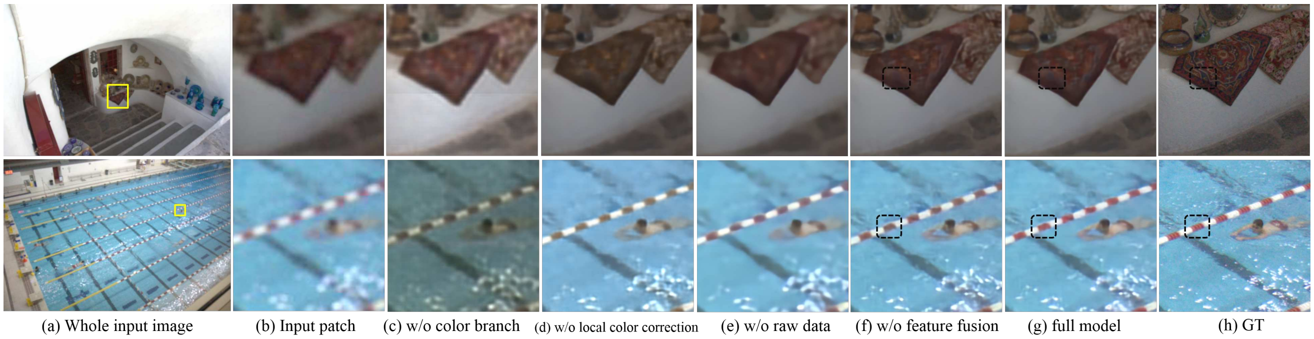


Figure 7. Ablation study of the proposed network on the synthetic dataset.

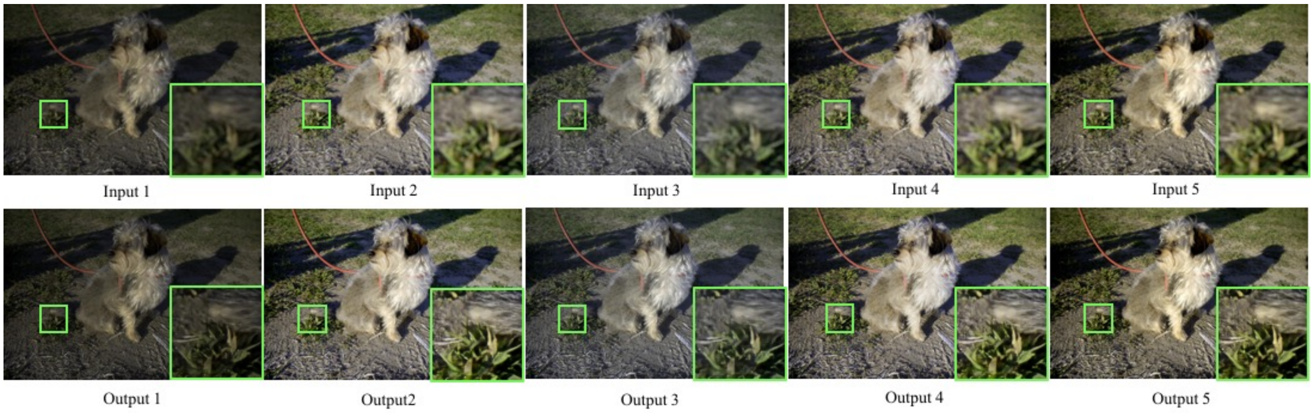


Figure 8. Generalization to manually retouched color images from MIT-Adobe 5K dataset [1]. The proposed method generates clear structures and high-fidelity colors.

Learning more complex color corrections. In addition to the reference images rendered by Dcraw, we also train the proposed model on manually retouched images by photography artists from the MIT-Adobe 5K dataset [1], which represent more diverse ISP systems. We test our model on the same raw input with different reference images produced by different artists in Figure 8. The proposed algorithm generates clear structures as well as high-fidelity colors, which shows that our method is able to generalize to more diverse and complex ISP systems.

4.3. Ablation Study

For better evaluation of the proposed algorithm, we test variant versions of our model by removing each component. We show the quantitative results on the synthetic datasets in Table 1 and provide qualitative comparisons in Figure 7. First, without the color correction branch, the network directly uses the low-resolution raw input for super-resolution, which cannot effectively recover high-fidelity colors for diverse cameras (Figure 7(c)). Second, simply adopting the global color correction strategy from [29] could only recover holistic color appearances, such as in the background of Figure 7(d), but there are still significant color errors in local regions without the proposed pixel-wise transformations. Third, to evaluate the importance of the raw input, we use the low-resolution color image as the input for both branches of our network, which is enabled by adaptively changing the packing layer of the image restoration branch. As shown in Figure 7(e), the model without raw input cannot generate fine structures and details due to the information loss in the ISP system. In addition, the network without feature fusion cannot predict accurate transformations and tends to bring color artifacts around subtle structures in Figure 7(f). By contrast, our full model effectively integrates different components, and generates sharper results with better details and fewer artifacts in Figure 7(g) by exploiting the complementary information in raw data and preprocessed color images.

4.4. Effectiveness on Real Images

We qualitatively evaluate our data generation method as well as the proposed network on real captured images. As shown in Figure 1(d) and 9(d), the results of RDN become sharper after re-training with the data generated by

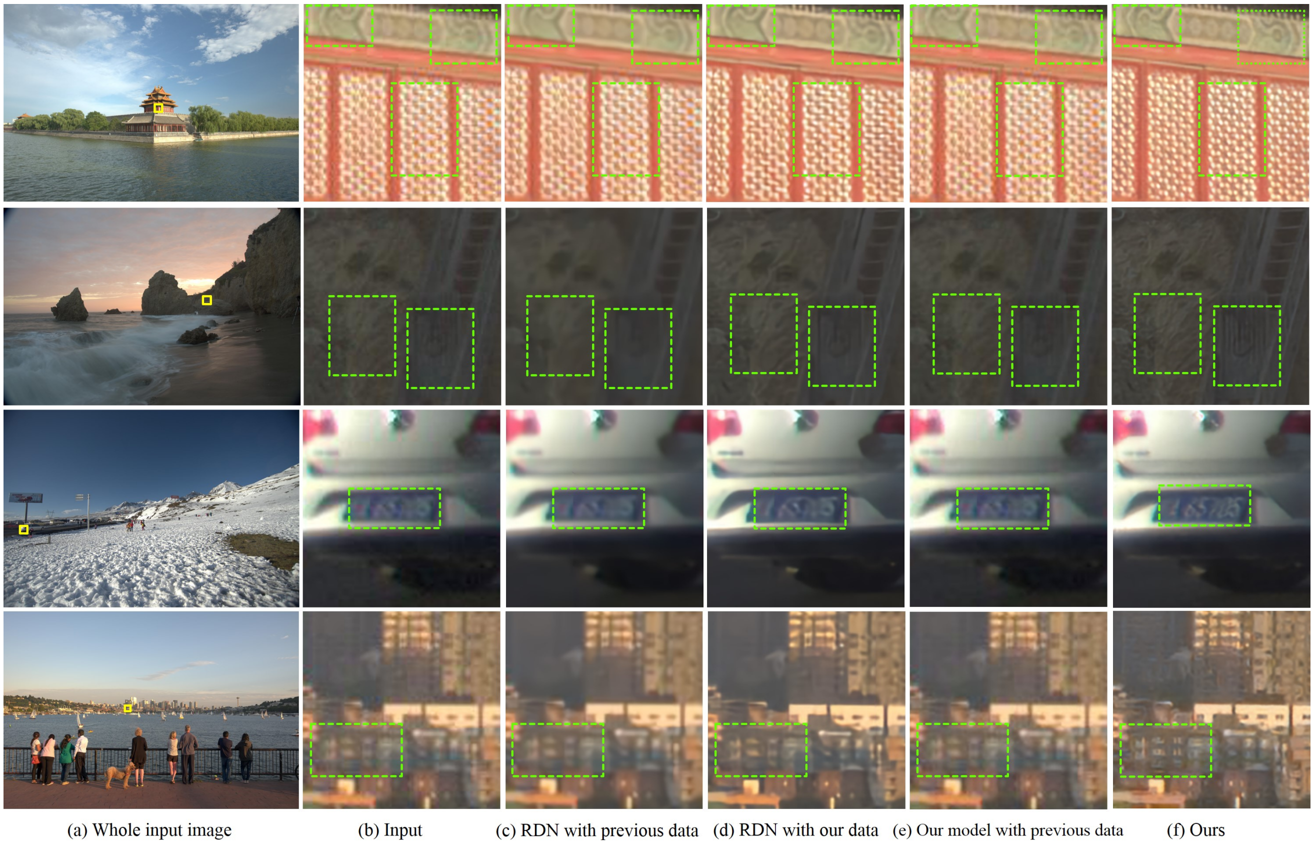


Figure 9. Results on real images. Since the outputs are of ultra-high resolution, typically 8000×12000 for a Leica camera and 4000×6000 for Canon, we only show image patches cropped from the tiny yellow boxes in (a). The images from top to bottom are captured by Canon, Sony, Nikon and Leica cameras, respectively.

our pipeline. On the other hand, the proposed dual CNN model cannot generate clear images (Figure 9(e)) by training on previous data generated by bicubic downsampling, homoscedastic Gaussian noise and non-linear space mosaicing [4, 9]. By contrast, we achieve better results with sharper edges and finer details by training the proposed network on our synthetic dataset as shown in Figure 1(e) and 9(f), which demonstrates the effectiveness of the proposed data generation pipeline as well as the raw super-resolution algorithm. Note that the real images are captured by different types of cameras, and all of them are not seen during training.

5. Conclusions

We propose a new pipeline to generate realistic training data by simulating the imaging process of digital cameras. In addition, we develop a dual CNN to exploit the radiance information recorded in raw data. The proposed algorithm compares favorably against state-of-the-art methods both quantitatively and qualitatively, and more importantly, enables super-resolution for real captured images. This work shows the superiority of learning with raw data, and we expect more applications of our work in other image processing problems.

References

[1] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, 2011. 2, 3, 5, 7

[2] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *CVPR*, 2018. 2, 3, 4, 6

[3] D. Coffin. Dcraw: Decoding raw digital photos in linux. http://www.cybercom.net/ dcoffin/dcraw/. 4

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 1, 2, 3, 5, 6, 8

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38:295–307, 2016. 2

[6] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 2

[7] S. Farsiu, M. Elad, and P. Milanfar. Multiframe demosaicing and super-resolution of color images. *TIP*, 15:141–159, 2006. 3

[8] Y. Gan, X. Xu, W. Sun, and L. Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *ECCV*, 2018. 5

[9] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 35:191, 2016. 3, 8

[10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 5

[11] J. E. Greivenkamp. *Field guide to geometrical optics*, volume 1. SPIE Press Bellingham, WA, 2004. 1

[12] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Trans. Graph. (SIGGRAPH Asia)*, 35(6), 2016. 2

[13] G. E. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *TIP*, 16:267–276, 1994. 4

[14] K. Hirakawa and T. W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *TIP*, 14:360–369, 2005. 4

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2, 4

[16] H. Jiang, Q. Tian, J. Farrell, and B. A. Wandell. Learning the image processing pipeline. *TIP*, 26:5032–5042, 2017. 3

[17] D. Khashabi, S. Nowozin, J. Jancsary, and A. W. Fitzgibbon. Joint demosaicing and denoising via learned nonparametric random fields. *TIP*, 23:4968–4981, 2014. 3

[18] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1, 2, 3, 5, 6

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[20] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate superresolution. In *CVPR*, 2017. 1, 2

[21] C. Liu, X. Xu, and Y. Zhang. Temporal attention network for action proposal. In *ICIP*, 2018. 1

[22] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 5

[23] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll. Burst denoising with kernel prediction networks. In *CVPR*, 2018. 4

[24] R. M. Nguyen and M. S. Brown. Raw image reconstruction using a self-contained srgb-jpeg image with only 64 kb overhead. In *CVPR*, 2016. 2

[25] W. B. Pennebaker and J. L. Mitchell. *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992. 2

[26] T. Plotz and S. Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 2, 4

[27] W. Ren, J. Zhang, X. Xu, L. Ma, X. Cao, G. Meng, and W. Liu. Deep video dehazing with semantic segmentation. *TIP*, 28:1895–1908, 2019. 1

[28] C. Schuler, H. Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *CVPR*, 2013. 4

[29] E. Schwartz, R. Giryes, and A. M. Bronstein. Deepisp: Towards learning an end-to-end image processing pipeline. *TIP*, 2018. 2, 3, 5, 6, 7

[30] 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 *CVPR*, 2016. 5

[31] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2014. 1

[32] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *ICCV*, 2017. 1, 2, 4, 5, 6

[33] P. Vandewalle, K. Krichane, D. Alleysson, and S. Süsstrunk. Joint demosaicing and super-resolution imaging from a set of unregistered aliased images. In *Digital Photography III*, 2007. 3

[34] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *ICCV*, 2015. 1, 2

[35] X. Xu, J. Pan, Y.-J. Zhang, and M.-H. Yang. Motion blur kernel estimation via deep learning. *TIP*, 27:194–205, 2018. 1

[36] X. Xu, D. Sun, S. Liu, W. Ren, Y.-J. Zhang, M.-H. Yang, and J. Sun. Rendering portraitures from monocular camera and beyond. In *ECCV*, 2018. 5

[37] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister, and M.-H. Yang. Learning to super-resolve blurry face and text images. In *ICCV*, 2017. 1, 2

[38] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 1, 2, 3, 5, 6

[39] R. Zhou, R. Achanta, and S. Süsstrunk. Deep residual network for joint demosaicing and super-resolution. *arXiv preprint arXiv:1802.06573*, 2018. 2, 3