Abstract—Image demosaicing and super-resolution are two important tasks in color imaging pipeline. So far they have been mostly independently studied in the open literature of deep learning; little is known about the potential benefit of formulating a joint demosaicing and super-resolution (JDSR) problem. In this paper, we propose an end-to-end optimization solution to the JDSR problem and demonstrate its practical significance in computational imaging. Our technical contributions are mainly two-fold. On network design, we have developed a Densely-connected Squeeze-and-Excitation Residual Network (DSERN) for JDSR. For the first time, we address the issue of spatio-spectral attention for color images and discuss how to achieve better information flow by smooth activation for JDSR. Experimental results have shown moderate PSNR/SSIM gain can be achieved by DSERN over previous naive network architectures. On perceptual optimization, we propose to leverage the latest ideas including relativistic discriminator and pre-excitation perceptual loss function to further improve the visual quality of reconstructed images. Our extensive experiment results have shown that Texture-enhanced Relativistic average Generative Adversarial Network (TRaGAN) can produce both subjectively more pleasant images and objectively lower perceptual distortion scores than standard GAN for JDSR. We have verified the benefit of JDSR to high-quality image reconstruction from real-world Bayer pattern collected by NASA Mars Curiosity.

Index Terms—Color imaging, Joint image demosaicing and super-resolution, deep learning, spatio-spectral attention.

I. INTRODUCTION

Image demosaicing and single image super-resolution (SISR) are two important image processing tasks to the pipeline of color imaging. Demosaicing is a necessary step to reconstruct full-resolution color images from so-called Color Filter Array (CFA) such as Bayer pattern. SISR is a cost-effective alternative to more expensive hardware-based solutions (i.e., optical zoom). Both problems have been extensively yet separately studied in the literature - from model-based methods [1], [2], [3], [4], [5], [6], [7], [8], [9] to learning-based approaches [10], [11], [12], [13], [14], [15], [16], [17], [18]. Treating demosaicing and SISR as two independent problems may generate undesirable edge blurring as shown in Fig. 1. Moreover, the processes of demosaicing and SISR can be integrated and optimized together from a practical application point of view (e.g., digital zoom for smartphone cameras such as Google Pixel 3 and Huawei P30).

Inspired by the success of joint demosaicing and denoising [20], we propose to study the problem of joint image demosaicing and super-resolution (JDSR) in this paper and develop a principled solution leveraging latest advances in deep learning to computational imaging. We argue that the newly formulated JDSR problem has high practical impact (e.g., to support the mission of NASA Mars Curiosity and smartphone applications). The problem of JDSR is intellectually appealing but has been under-researched so far. The only existing work we can find in the open literature is a recently published paper [21] which contained a straightforward application of ResNet [22] and considered the scaling ratio of two only. As demonstrated in Fig. 1 our optimized solution to JDSR can achieve significantly better visual quality (both subjectively and objectively) than the ad-hoc approach [21].

The motivation behind our approach is mainly two-fold. On one hand, rapid advances in deep residual learning have offered a rich set of tools for image demosaicing and SISR. For example, DenseNet [23] has been adapted to fully exploit hierarchical features for the problem of SR in SRDenseNet [24] and residual dense network (RDN) [17]; residual channel attention network (RCAN) [18] allows us to develop much deeper networks (over 1000 layers) with squeeze-and-excitation (SE) blocks [25] than previous works (e.g., [14], [16]). However, to the best of our knowledge, the issue of spatio-spectral attention mechanism has not been explicitly addressed for color images in the open literature. How to jointly exploit spatial and spectral dependency for JDSR in network design deserves a systematic study.

On the other hand, we propose to optimize the perceptual quality for JDSR because that is what really matters in real-world applications. Generative adversarial network [27] is arguably the most popular approach toward perceptual...
optimization and has demonstrated convincing improvement for SISR in SRGAN [15]. However, it has also been widely observed that the training of GAN suffers from stability issue which could have catastrophic impact on reconstructed images. There has been a flurry of latest works (e.g., Relativistic average GAN (RaGAN) [28], enhanced SRGAN (ESRGAN) [16] and perception-enhances SR (PESR) [29]) showing the potential of relativistic discriminator in stabilizing GAN and improving visual quality of SISR images. How to leverage those latest advances to optimize the perceptual quality for JDSR has practical significance.

Overall, our contributions are summarized as follows:

- **Network design**: we propose a Densely-connected Squeeze-and-Excitation Residual Network (DSERN) for JDSR. A novel Dense Squeeze-and-Excitation Residual Block (D\_SERB) is designed to facilitate information flow in deeper and wider networks by smooth activation, which can more effectively suppress spatio-spectral aliasing.

- **Perceptual optimization**: we have leveraged the latest advance RaGAN [28] from SISR to JDSR and studied the choices of perceptual loss function for JDSR. In addition to improved stability, we have found that Texture-enhanced RaGAN (TRaGAN) with a before-activation perceptual loss function can produce visually more pleasant results.

- **Real-world application**: we have applied the proposed DSERN+TRaGAN solution to raw Bayer pattern data collected by the Mast Camera (Mastcam) of NASA Mars Curiosity Rover. Our experimental results have shown visually superior high-resolution image reconstruction can be achieved at the scaling ratio as large as 4.

II. RELATED WORKS

Both image demosaicing and super-resolution have been studied in decades in the open literature. In this section, we review image demosaicing and image super-resolution approaches separately and focus on deep learning based methods.

A. Image Demosaicing

Existing approaches toward image demosaicing can be classified into two categories: model-based methods [1, 2, 3, 4] and learning-based methods [10, 11, 13]. Model-based approaches rely on hand-crafted parametric models which often suffer from lacking of the generalization capability to handle varying characteristics in color images (i.e., the potential model-data mismatch). Recently, deep learning methods show the advantages in image demosaicing field. Inspired by single image super-resolution model SRCNN [30], DMCNN [31] utilized super-resolution based CNN model and ResNet [22] to investigate image demosacing problem. CDMCNN [32] introduced to apply residual learning [22] with a two-phase network architecture which firstly recovers green channel as guidance prior and then uses this guidance prior to reconstruct the RGB channels. Besides to explore image demosacing methods only, there are several works studying joint image demosaicing and denoising (JDD) problem. Dong et.al. [33] developed a deep neural network with generative adversarial networks (GAN) [27] and perceptual loss functions to solve JDD problems. Inspired by classical image regularization and majorization-minimization optimization, Kokkinos and Lefkimmiatis [12] proposed a deep neural network to solve JDD problem. Deep learning based image demosaicing techniques have shown convincingly improved performance over model-based ones on several widely-used benchmark dataset (e.g., Kodak and McMaster [20]). However, the issue of suppressing spatio-spectral alising has not been addressed in the open literature as far as we know.

![Fig. 2. Overview of proposed DSERN network architecture, the generator is shown in (a), D\_RG stands for Dense Residual Group and \oplus denotes element-wise sum; (b) is the structure of discriminator used (s is the stride and c is the number of feature maps).](image)
B. Image Super-resolution

Model-based approaches towards SISR [5, 6, 7, 8, 9] suffer from notorious aliasing artifacts and edge blurring. Recently, deep learning-based approaches have advanced rapidly. SRCNN [30] first introduced deep learning based method to solve single image super-solution task with three convolutional layers and achieved much better performance than model based methods. Benefit by concept of ResNet [22], VDSR [14] firstly trained 20 layers deep networks with long residual connection which can only learn more high-frequency information and increase the convergence speed. EDSR [26] proposed to integrate several resblocks and remove batch-normalization layer, which can save GPU memory, stack more layers and make networks wider, to further improve SISR performance. LapSRN [34] proposed to super-resolve LR image several times to save GPU memory and achieve better performance.

Most recent advances include SRDenseNet [24] which applied denseNet [23] to solve SISR task, RDN [17] which utilized ResNet and DenseNet to create residual dense block (RDB). Through local feature fusion, the proposed RDB can allow larger growth rate to boost the performance. RCAN [18] first introduced attention mechanism inspired by SENet [25] to calibrate feature maps and proposed residual in residual structure to achieve a very deep convolutional networks which achieved new state-of-art performance for SISR task. Besides objective measures such as PSNR/SSIM [35], SRGAN [13] introduced a novel generative adversarial networks (GAN) based architecture to optimize the perceptual quality of SR images, benefit by GAN, SRGAN can reconstruct more textures from low-res images. An enhanced version of SRGAN named ESRGAN [16] using relativistic average GAN (RaGAN) was developed in [28] as well as [29] which can recover more realistic super-resolved image compared with SRGAN. By contrast, the problem of JDSR has been under-researched so far with the only exception of [21].

III. NETWORK DESIGN: SPATIO-SPECTRAL ATTENTION

The hierarchy of our network design goes like: DSERN (Fig. 2) → D_RG subnetwork (Fig. 3a) → D_SERB module (Fig. 3b) → DSE block (Fig. 4).

A. DSERN: Deeper and Wider are Better

Channel attention mechanism has been successfully applied in both high-level (e.g., SENet [25] and LS-CNN [30]) and low-level (e.g., RCAN [18]) tasks. A channel attention module first squeezes the input feature map and then activates one-time reduction-and-expansion to excite the squeezed feature map. Such strategy is not optimal for recovering missing high-frequency information in SISR when the network is very deep; meanwhile, JDSR problem requires simultaneous recovery of incomplete color information across R, G, B bands, which requires extra attention toward the dependency among spectral bands. How to generalize the channel attention mechanism from spatial-only to joint spatio-spectral serves as the key motivation behind our approach.

As discussed in [18], high-frequency components often correspond to regions in an image such as textures, edges, corners and so on. Conventional Conv layers have limited capability of exploiting contextual information outside the local receptive field especially due to missing data in Bayer pattern. To overcome this difficulty, we propose to design a new Densely-Connected Squeeze-and-Excitation Residual Block (D_SERB) as shown in Fig. 3b and Fig. 4. The proposed D_SERB is designed to implement a deeper and wider spatio-spectral channel attention mechanism for the purpose of more effectively suppressing spatio-spectral aliasing in low-resolution (LR) Bayer pattern.

Unlike SENet [25] and RCAN [18] (using one-time reduction-and-expansion), D_SERB uses multiple expansion modules after reduction to assure more faithful information recovery when the network gets deeper and wider. As shown in Fig. 3 we have kept both long skip and short skip connections like RCAN in order to make the overall training stable and facilitate the information flow both inside and outside the D_SERB modules. Although similar idea of local feature fusion existed in residual dense block of RDN [17], our hybrid design - i.e., the Dense SE (DSE) block combining the ideas in RDN and RCAN - is novel from the perspective of achieving joint spatio-spectral attention for JDSR. Spatio-spectral channel attention mechanism in the proposed D_SERB module can help to recalibrate input features via channel statistics across spectral bands. In SISR, one-time reduction-and-expansion operation might be sufficient for capturing channel-wise dependencies for low-resolution color images; however our JDSR task aims at recovering two-third of missing data in spectral bands in addition to high-frequency spatial details, which calls for the design of deeper and wider networks.
B. Dense Squeeze-and-Excitation Residual Block

The key to deeper and wider networks lies in the design of D_SERB module - i.e., how to use multiple expansions after reduction to assure more faithful information recovery both inside and outside D_SERB modules? As shown in Fig. 3(b), we propose a Dense Squeeze-and-Excitation (DSE) block in which the channel size can be expanded step by step (see Fig. 4). The key advantages of this newly designed DSE block include: 1) the reduced channel descriptor can be smoothly activated multiple times and therefore more faithful information across spatio-spectral domain is accumulated; 2) dense-connection can increase the network depth and width without running into the notorious vanishing-gradient problem [57]; 3) both information flow and network stability, which are important to a principled solution to JDSR, can be jointly improved by introducing dense connections to SE residual blocks (so we can train even deeper than RCAN [18]).

More specifically, to implement the DSE block, we first apply global average pooling to squeeze input feature maps. Let us denote the input feature maps by \( U = [u_1, u_2, ..., u_C] \), which contains \( C \) feature maps with the dimension of \( H \times W \). Then the global average pooling output \( z \in \mathbb{R}^C \) can be calculated by:

\[
z_C = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_C(i,j)
\]

where \( z_C \) is the \( C \)-th element of \( z \), \( u_C(i,j) \) is the pixel value of the \( C \)-th feature at position \((i,j)\) from input feature maps. Then we propose to implement a simple gating mechanism as adopted by previous works including SENet [25] and RCAN [18]:

\[
s = \sigma(W_E(\delta(W_S(z))))
\]

where \( \sigma \) refers to a sigmoid function, \( \delta \) denotes the ReLU function. Note that both \( W_S \) and \( W_E \) are Conv layers with weights \( W_S \in \mathbb{R}^{1 \times 1 \times \gamma} \) and \( W_E \in \mathbb{R}^{1 \times 1 \times C} \), \( r \) is the reduction ratio to reduce the dimension of \( z \) (details about this hyperparameter controlling the tradeoff between the capacity and the complexity can be found in SENet [25]).

In order to achieve deeper and wider channel attention, we propose a novel strategy of activating the reduced features step by step (instead of one-shot) with dense connections. As shown in Fig. 4 after reducing \( z \) by a factor of \( r \), we can gradually expand (i.e., smooth activation) the feature map \( N \) times where \( N = r - 1 \). The detailed procedure of our proposed DCA module can be written as:

\[
X_0 = \delta(W_S(z)), \quad X_i = \delta(W_E(\sigma(W_S([X_0, X_1, X_2, ..., X_N]))) \quad (i \in [1, N])
\]

\[
\hat{s} = \sigma(W_E(\delta(W_S(z))))
\]

where \( \sigma, \delta, W_E, W_S \) are the same as Eq. (2), \( W_S \in \mathbb{R}^{1 \times 1 \times \gamma} \) and \( [X_0, X_1, X_2, ..., X_N] \) refers to the concatenation of feature maps by each DSE layer.

Finally, we can rescale the input feature map by

\[
\hat{U} = \hat{s} \cdot U
\]

With the new DSE block, we can train even deeper network than RCAN [18] thanks to the improved information flow.

IV. PERCEPTUAL OPTIMIZATION: RELATIVISTIC DISCRIMINATOR AND LOSS FUNCTION

A. Texture-enhanced Relativistic average GAN (TRaGAN)

The discriminator \( D \) in standard GAN [27] only estimates the probabilities of real/fake images, and the interaction between generator and discriminator is interpreted as a two-player minimax game. It can be expressed as \( D(x) = \sigma(C(x)) \), where \( \sigma \) is sigmoid function, \( C(x) \) is non-transformed layer, \( x \) is the input image. Such idea has been successfully applied to the problem of SISR such as SRGAN [15] in which the super-resolved image (fake version) is compared against the ground-truth (real version). In other words, discriminator \( D \) serves as a judge for perceptual optimization of generator (as shown in Fig. 2(b)).

Unlike standard GAN, relativistic average GAN (RaGAN) [28] can make the discriminator \( D \) to estimate the probability based on both real and fake images, making a real image more realistic than a fake one (on the average). According to [28], RaGAN can not only generate more realistic images but also stabilize the training progress. Recently, the benefit of RaGAN over conventional GAN has been demonstrated for SISR in [16] and [29]. Here we propose to leverage the idea of RaGAN to JDSR and demonstrate how relativistic discriminator can work with the proposed DSERN (generator) for the purpose of perceptual optimization (overlooked in RDN [17] and RCAN [18]).

To implement RaGAN, we represent the real and fake images by \( x_r \) and \( x_f \) respectively; then we can formulate the output of a modified discriminator \( \hat{D} \) for RaGAN by:

\[
\hat{D}(x_r) = \sigma(C(x_r) - E_{x_f}[C(x_f)]) \quad (7)
\]

\[
\hat{D}(x_f) = \sigma(C(x_f) - E_{x_r}[C(x_r)]) \quad (8)
\]

where \( E_{x_f} \) and \( E_{x_r} \) are the expectation functions. It follows that the discriminator loss function \( L_{D_{RaGAN}} \) and adversarial loss function \( L_{G_{RaGAN}} \) can be written as:

\[
L_{D_{RaGAN}} = -E_{x_r} [\log(\hat{D}(x_r))] - E_{x_f} [\log(1 - \hat{D}(x_f))] \quad (9)
\]

\[
L_{G_{RaGAN}} = -E_{x_r} [\log(1 - \hat{D}(x_r))] - E_{x_f} [\log(\hat{D}(x_f))] \quad (10)
\]

It has been observed that the class of texture images is often more difficult for SISR due to spatial aliasing [29]. One way of achieving better texture reconstruction is through attention mechanism at the image level - i.e., to emphasize (i.e., increase the weight) difficult samples and overlook (i.e., down-weighting) easy ones. Such idea of weighting can be conveniently incorporated into the RaGAN package because the PyTorch implementation allows an optional weight input. More specifically, we propose to consider the following weighted function with a new hyperparameter \( \gamma \) tailored for Texture enhancement:

\[
L_{G_{RaGAN}} = - \sum_i (\hat{D}(x_r))^\gamma \log(1 - \hat{D}(x_r))
\]

\[
- \sum_i (1 - \hat{D}(x_f))^\gamma \log(\hat{D}(x_f))
\]
B. Perceptual Loss Function

We have implemented the following perceptual loss function based on [38, 15, 16, 19]. With a pre-trained VGG19 model [39], we can extract high-level perceptual features of both high-resolution (HR) and SR images from the 4-th convolutional layer of VGG19 before the activation function is applied. Inspired by [16], we propose to extract high-level features before the activation function layer because it can further improve the performance. Let’s define perceptual loss as $L_{vgg}$ and $L_1$-norm distance as $L_1$. Then the total loss for our generator $L_G$ can be formulated as follows:

$$L_G = L_{vgg} + \lambda_1 L_{G\text{RagGAN}}^T + \lambda_2 L_1 \quad (12)$$

where coefficients $\lambda_1$ and $\lambda_2$ are used to balance different loss terms. $L_{vgg} = \Phi(f(SR), f(HR))$. $\Phi$ denotes the mean-squared error function (MSE), $f(SR)$ and $f(HR)$ are the high-level features extracted from VGG19-54 layer. Note that although similar loss functions were considered in previous studies including [16] and [29], their experiments include synthetic low-resolution images only. In this paper, we will demonstrate the effectiveness of the proposed perceptual optimization for JDSR on real-world data next.

V. Experimental results

A. Implementation details

In our proposed DSERN networks, we have kept the basic setting same as RCAN [18]: D_RG is set to 10 and every D_RG contains 20 D_SERBs. All kernel size of Conv layers is $3 \times 3$ with 64 filters ($C = 64$) except the Conv layers in our DSE modules. The reduction ratio is $r = 16$. The upscale module we have used is the same as [40]. The last layer filter is set to 3 in order to output super-resolved color images. For the discriminator setting, we have implemented the same discriminator networks structure as SRGAN [15]. All kernel size of Conv layers is $3 \times 3$ as shown in Fig. 2g).

In our PyTorch implementation of DSERN, we first randomly crop the 3-channel Bayer patterns as small patches with the size of $48 \times 48$, and crop the corresponding HR color images, with a batch size of 16; then we augment the training set by standard geometric transformations (flipping and rotation). Our model is trained and optimized by ADAM [41] with $\beta_1 = 10^{-3}$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The initial learning rate is set to $1 \times 10^{-4}$, the decay factor is set to 5, which decreases the learning rate by half after [80k, 120k, 150k, 180k] steps; the $L_1$ loss function is applied to minimize the error between HR and SR images. To train GAN-based networks, we have used the trained DSERN to initialize the generator of GAN to get a better initial SR image for discriminator. The same learning rate and decay strategies are adopted here. $\lambda_1$ and $\lambda_2$ in Eq. (12) are set to $5 \times 10^{-3}$ and $1 \times 10^{-2}$ respectively as [16].

Because the codes of RDSR [21] are not publicly available, we have tried our own best to reproduce RDSR using PyTorch while keeping the batch size (16), patch size ($64 \times 64$) and number of residual blocks (24) the same as the original work [21]. The learning rate and decay steps in RDSR implementation are the same as those in our DSERN. This way, we have striven to make the experimental comparison against RDSR [21] as fair as possible.

B. Training Dataset

In our experiment, we have used DIV2K dataset [42] as the training set, which includes 800 images (2K resolution). For testing, we have evaluated both popular image super-resolution benchmark datasets including Set5 [5], Set14 [43], B100 [44], Urban100 [45], and Manga109 [46], and popular image demosaicing datasets such as McMaster [47] and Kodak PhotoCD. To pre-process training and testing data, we downsample original high-resolution images by a factor of $2 \times, 3 \times, 4 \times$ using Bicubic interpolation then generate the ‘RGGB’ Bayer pattern. Based on previous work [31] and our own study (refer to next paragraph), supplying three-channels separately as the input (instead of the mosaicked single-channel composition) works better for the proposed network architecture. All experiments are implemented using PyTorch framework [48] and trained on NVIDIA Titan Xp GPUs.

Note that we have to be careful about four different spatial arrangements of Bayer patterns [49] in our definition of
TABLE II
OBJECTIVE PERFORMANCE COMPARISON AMONG DIFFERENT METHODS IN TERMS OF PERCEPTUAL INDEX (THE LOWER THE BETTER). BOLT INDICATES THE BEST RESULT AND UNDERLINE THE SECOND BEST.

| Methods                  | Scale | Set5 | Set14 | B100 | Urban100 | Manga109 | McM | PhotoCD |
|--------------------------|-------|------|-------|------|----------|----------|-----|---------|
| FlexISP [19]+RCAN [18]   | x2    | 4.16 | 4.14  | 3.34 | 4.04     | 4.97     | 3.51| 5.42    |
| RDSR [21]                | x2    | 4.15 | 3.81  | 3.31 | 4.08     | 3.96     | 3.28| 5.61    |
| DSERN (ours)             | x2    | 4.17 | 3.81  | 3.28 | 4.09     | 4.07     | 3.27| 5.65    |
| DSERN_GAN (ours)         | x2    | 3.42 | 3.03  | 2.40 | 3.59     | 3.58     | 2.96| 4.94    |
| DSERN_TRaGAN (ours)      | x2    | 3.11 | 2.84  | 2.33 | 3.56     | 3.44     | 2.40| 4.73    |

| FlexISP [19]+RCAN [18]   | x3    | 6.98 | 5.70  | 6.18 | 5.55     | 5.43     | 5.14| 6.42    |
| RDSR [21]                | x3    | 5.57 | 4.59  | 4.60 | 4.69     | 4.39     | 4.56| 6.43    |
| DSERN (ours)             | x3    | 5.71 | 4.74  | 4.48 | 4.75     | 4.53     | 4.57| 6.52    |
| DSERN_GAN (ours)         | x3    | 4.10 | 3.18  | 2.51 | 3.68     | 3.61     | 2.58| 5.33    |
| DSERN_TRaGAN (ours)      | x3    | 3.58 | 2.89  | 2.33 | 3.59     | 3.41     | 2.42| 4.76    |

| FlexISP [19]+RCAN [18]   | x4    | 7.42 | 6.63  | 6.30 | 5.55     | 5.28     | 7.15| 6.88    |
| RDSR [21]                | x4    | 6.36 | 5.95  | 5.98 | 5.30     | 4.94     | 5.69| 6.78    |
| DSERN (ours)             | x4    | 6.18 | 5.94  | 5.92 | 5.27     | 5.00     | 5.68| 6.87    |
| DSERN_GAN (ours)         | x4    | 3.95 | 2.96  | 2.46 | 3.57     | 3.45     | 2.61| 4.75    |
| DSERN_TRaGAN (ours)      | x4    | 3.72 | 2.94  | 2.40 | 3.51     | 3.51     | 2.54| 4.65    |

**C. PSNR/SSIM Comparisons**

It is convenient to further improve the performance of our DSERN by a so-called self-ensemble strategy (as done in previous works [20, 50, 18, 17]). The improved results are denoted as “DSERN+”. We have compared our methods against two benchmark methods: a separated (brute-force) approach Flex [19] + RCAN [18], recently published literature RDSR [21]. To evaluate the results of Flex [19] + RCAN [18] approach, we first demosaiced the LR mosaiced images by using Flex to get LR color images, then super-resolved them by applying a pre-trained RCAN model. Note that we have used the pre-trained RCAN weights provided by the authors on GitHub.

Table I shows PSNR/SSIM comparison results for scaling factors of $2 \times$, $3 \times$ and $4 \times$. It can be seen that our DSERN+ method perform the best for all datasets and scale factors. Even without self-ensemble, the performance of DSERN still leads all of datasets and scaling factors. We observe that moderate PSNR/SSIM gains ($0.2–0.5dB$) over previous state-of-the-art. Since PSNR/SSIM metrics do not always faithfully reflect the visual quality of images, we have also included the subjective quality comparison results for image “TotteokiNoABC” in Fig. 9. It can be readily observed that for the top of the pink sock, only our DSERN can faithfully recover stripe patterns; both brute-force approach (Flex+RCAN) and RDSR have produced severe blurring artifacts. Taking another example, Fig. 7 shows the comparison at two other scaling factors ($3 \times$ and $4 \times$). For “img_062”, we observe that all approaches contain noticeable visual distortion, but our DSERN method can recover more shape details than other competing approaches; for “253027”, zebra pattern recovered by DSERN appears to have the highest quality. For more visual comparison, see Fig. 9 and Fig. 10 which show more visual comparison among various competing approaches (please zoom in for a detailed comparison).

**D. Perceptual Index (PI) Comparisons**

Most recently, a new objective metric called Perceptual Index (PI) [51] has been developed for perceptual SISR (e.g., the 2018 PIRM Challenge [52]). The PI score is defined by

$$PI = \frac{1}{2}((10 - MA) + NIQE)$$

where MA denotes a no-reference quality metric [53] and NIQE referred to Natural Image Quality Evaluator [54]. Note that the lower PI score, the better perceptual quality (i.e., contrary to SSIM [55]). Objective comparison of competing feature maps. One can either treat the Bayer pattern like a gray-scale image (one-channel setting) which ignores the important spatial arrangement of R/G/B; or take spatial arrangement as a priori knowledge and pad missing values across R,G,B bands by zeroes (three-channel setting). As shown in Fig. 5, the former has the tendency of producing color misregistration artifacts, which suggests the latter works better. Our experimental result has confirmed a similar finding previously reported in [31].
JDSR methods in terms of PI is shown in Table II. We have observed that GAN-based methods produce the lowest PI scores for all datasets and scaling factors. Fig. 8 provides the visual comparison with image “IMG0019” (4×). It can be observed that GAN-based methods can recover sharper edges and overcome the issue of over-smoothed regions. Additionally, TRaGAN is capable of achieving even lower PI scores than standard GAN.

E. Ablation Studies

To demonstrate the effect of proposed DSE module, we study the networks: 1) only based on ResNet; 2) ResNet with channel attention module (RCAN); 3) ResNet with proposed DSE module (DSERN). All three networks are trained under same setting for fair comparison. The general SR benchmark datasets are used, scale factor is 2. From Table III we have found that ResNet has similar performance on Set5, Set14 and B100 to more advanced RCAN and DSERN. But when compared on Urban100, Manga109 and McM, RCAN and DSERN have better performance than ResNet; and the proposed DSERN has the best performance on most benchmark datasets.

F. Performance on the Real-world Data

Finally, we have tested our proposed JDSR technique on some real-world data collected by the Mastcam of NASA Mars Curiosity. The raw data are ‘RGGB’ bayer pattern sized by 1600 × 1200. Due to hardware constraints, the left camera and the right camera of Mastcam have different focal lengths (the left is about 3 times weaker than the right). To compensate such a “lazy-eye” effect on raw Bayer patterns, it is desirable to develop a joint demosaicking and SR technique with at least a scaling factor of 3 (in order to support high-level standard stereo-based vision tasks such as 3D reconstruction and object recognition). Our proposed JDSR algorithm is a perfect fit for this task, which shows the great potential of computer vision and deep learning in deep space exploration.

The visual comparison results are shown in Fig. 11 for a scaling factor of 4. It can be seen that brute-force ap-
Table III

| Method          | Scale | Set5       | Set14       | B100       | Urban100   | Manga109   | McM        |
|-----------------|-------|------------|------------|------------|------------|------------|------------|
| ResNet          | x2    | 36.48/0.9498 | 32.71/0.9030 | 31.67/0.8876 | 31.65/0.9201 | 36.48/0.9642 | 36.11/0.9443 |
| RCAN            | x2    | 36.54/0.9499 | 32.74/0.9032 | 31.68/0.8878 | 31.74/0.9200 | 36.65/0.9643 | 36.18/0.9445 |
| DSERN (ours)    | x2    | **36.55/0.9500** | 32.71/0.9031 | **31.70/0.8879** | **31.78/0.9207** | **36.72/0.9652** | **36.23/0.9448** |

TABLE III

Ablation study for ResNet, ResNet with CA (RCAN) and ResNet with proposed DSERN. Bold font indicates the best result and underline the second best.

For approach (Flex+RCAN) suffers from undesired artifacts especially around the edge of rocks. Our proposed DSERN method can overcome this difficulty but the result appears over-smoothed. DSERN_GAN improves the visual quality to some degree - e.g., more fine details are present and sharper edges can be observed. Replacing GAN by TRaGAN can further improve the visual quality not only around the textured regions (e.g., roads and rocks) but also in the background (e.g., terrain appears visually clearer and sharper). Fig. 12 shows the visual comparison among Flex+RCAN, DSERN, DSERN_GAN and DSERN_TRaGAN approaches. The raw image is captured by the right eye of NASA Mast Camera. The scale factor is 4 (please zoom in to get a better view).

VI. Conclusion

In this paper, we proposed to study the problem of joint demosaicing and super-resolution (JDSR) - a topic has been underexplored in the literature of deep learning. Our solution consists of a new densely-connected squeeze-and-excitation residual network for image reconstruction and an improved GAN with relativistic discriminator and new loss functions for texture enhancement. Compared with naive network designs,
Fig. 10. Visual quality comparison of JDSR results among competing approaches at a scaling factor of 3 or 4.

ACKNOWLEDGMENT

The authors would like to thank Dr. Chiman Kwan for supplying real-world Bayer pattern collected by NASA Mars Curiosity. This work is partially supported by the DoJ/NIJ under grant NIJ 2018-75-CX-0032, NSF under grant OAC-1839909 and the WV Higher Education Policy Commission Grant (HEPC.dsr.18.5).

REFERENCES

[1] 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, 2005.

[2] X. Li, B. Gunturk, and L. Zhang, “Image demosaicing: A systematic survey,” in *Visual Communications and Image Processing* 2008, vol. 6822. International Society for Optics and Photonics, 2008, p. 68221J.

[3] X. Li, “Demosaicing by successive approximation,” *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.
Fig. 11. Visual quality comparison of JDSR results on real-world Bayer pattern collected by NASA Mars Curiosity (4×).

Fig. 12. More visual quality comparison of JDSR results on real-world Bayer pattern collected by NASA Mars Curiosity.
[50] R. Timofte, R. Rothe, and L. Van Gool, “Seven ways to improve example-based single image super resolution,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1865–1873.

[51] Y. Blau and T. Michaeli, “The perception-distortion tradeoff,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[52] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, “2018 PIRM Challenge on Perceptual Image Super-resolution,” arXiv preprint arXiv:1809.07517, 2018.

[53] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, “Learning a no-reference quality metric for single-image super-resolution,” Computer Vision and Image Understanding, vol. 158, pp. 1–16, 2017.

[54] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a” completely blind” image quality analyzer.” IEEE Signal Process. Lett., vol. 20, no. 3, pp. 209–212, 2013.