ABSTRACT

Recently, the single image super-resolution (SISR) approaches with deep and complex convolutional neural network structures have achieved promising performance. However, those methods improve the performance at the cost of higher memory consumption, which is difficult to be applied for some mobile devices with limited storage and computing resources. To solve this problem, we present a lightweight multi-scale feature interaction network (MSFIN).

For lightweight SISR, MSFIN expands the receptive field and adequately exploits the informative features of the low-resolution observed images from various scales and interactive connections. In addition, we design a lightweight recurrent residual channel attention block (RRCAB) so that the network can benefit from the channel attention mechanism while being sufficiently lightweight. Extensive experiments on some benchmarks have confirmed that our proposed MSFIN can achieve comparable performance against the state-of-the-arts with a more lightweight model.

Index Terms—Image super-resolution, Lightweight, Multi-scale interaction network, Recurrent channel attention.

1. INTRODUCTION

The purpose of single image super-resolution (SISR) is to attain a high-resolution (HR) image from its degraded low-resolution (LR) counterpart. For this task, many super-resolution (SR) methods have been designed and shown prominent performance, including early reconstruction-based, and more recently learning-based methods.

Recently, the solutions based on convolution neural network (CNN) have shown excellent performance in SISR task. Dong et al. [1] first presented a three-layer convolutional neural network (SRCNN) for image SR in an end-to-end manner, and achieved prominent results against previous work.

Since then, many deep super-resolution networks have been proposed, such as VDSR [2], MemNet [3], LapSRN [4], and MSRN [5]. Nowadays, the attention mechanism has been widely utilized in computer vision tasks. Hu et al. [6] proposed the squeeze-and-excitation network (SENet) and yielded significant performance improvement in image classification task by utilizing the relationship between channels. Zhang et al. [7] presented the so-called residual channel attention network (RCAN) by using some residual channel attention blocks (RCAB) for image SR tasks.

However, while these complex and deep SR models can provide significant performance improvement, they can be difficult to be used to the real-world scenarios on account of the large number of parameters and computation. Especially for some mobile devices, their storage and computing resources are limited. Many works have shown that recursive networks can effectively reduce the number of parameters. For example, DRCN [8] and DRRN [9] adopted recursive mechanism for parameter sharing. However, simply using recursion can achieve parameter reduction but at the cost of performance degradation. To handle this problem, an optional solution is to design some lightweight and effective SR models. Ahn et al. [10] proposed a cascading residuals network for mobile scenarios (CARN-M), which reduces the computation at the cost of PSNR reduction. Hui et al. [11] designed an information distillation network (IDN) that used a channel split strategy to aggregate current information with locally retained information. Then, on the basis of IDN, they further proposed an efficient information multi-distillation network and adaptive cropping strategy to further enhance the performance [12]. Lan et al. [13] proposed a residual multi-scale module based on attention mechanism (MADNet) to strengthen the feature representation ability of the model. Similarly, Li et al. [14] develop a lightweight super-resolution network (s-LWSR) for efficient SISR.

For those aforementioned methods, there are still some issues to be addressed. First, it is found that RCAB generated a large number of parameters, which are mainly from the convolutional layers in the residual module. In addition, in multi-scale features based methods, the informative features
from different scales are separately aggregated to generate the final HR features, the mutual collaboration between them are not fully exploited. Based on these observations, in order to exploit image features and restore more details, we present a multi-scale feature interaction network (MSFIN), which can make a better trade-off between performance and the number of parameters. In summary, our major contributions can be summarized as follows: (i) Based on the channel attention mechanism, we propose a lightweight recurrent channel attention block (RRCAB) for lightweight image SR task, which can improve the quality of reconstructed images with smaller memory consumption. (ii) Based on these RRCABs, we develop a multi-scale feature interaction module, which can make full use of the informative features from various scales and interactive connection to better restore the details of the generated images.

2. PROPOSED METHOD

2.1. Network Architecture

In this part, we describe our proposed multi-scale feature interaction network (MSFIN) in detail. As shown in Fig.1, our MSFIN is composed of three parts: the shallow feature extraction part, the multi-scale feature extraction part and the reconstruction part. In our works, the original LR image $I_{LR}$ is first preprocessed by the interpolation algorithm and then upsampled to the size of the target HR image. Specifically, similar to some recent super-resolution networks, a $3 \times 3$ convolution layer is used in the shallow feature extraction part:

$$F_{SF} = H_{sext}(I_{LR}),$$

where $H_{sext}$ denotes the shallow feature extraction function and $F_{SF}$ are the extracted features to be sent to the multi-scale feature extraction part for deep feature learning.

The multi-scale feature extraction part is composed of deep feature extraction module and multi-scale feature interaction module (MSFIM). The multi-scale feature extraction part can be formulated as:

$$F_{DF} = H_{msfe}(F_{SF}) = H_{dext}(H_{msfim}(F_{SF}))$$

where $H_{msfe}$ denotes the multi-scale feature extraction function, $H_{dext}$ and $H_{msfim}$ represent deep feature extraction function module and MSFIM respectively. $F_{DF}$ denotes the extracted deep features. The reconstruction part can be formulated as

$$I_{SR} = H_{re}(F_{DF}) = H_{MSFIN}(I_{LR}).$$

where $H_{re}$ denotes the reconstruction function, $H_{MSFIN}$ denote the SR network, and $I_{SR}$ is the reconstructed image.

The network will be optimized with the $L_1$ loss function. Given a training set $\{I_{jLR}, I_{jHR}\}_{j=1}^M$, where $M$ denotes the number of training patches, the loss function of our MSFIN with the parameter set $\Theta$ can be formulated as

$$L(\Theta) = \frac{1}{M} \sum_{i=1}^M ||H_{MSFIN}(I_{jLR}) - I_{jHR}||_1.$$
consists of several lightweight recurrent residual channel attention blocks (RRCAB). The L1 level can be formulated as

\[ F_{L1}^2 = H_{L1}^1(F_{SF} + H_{deco}(F_{L2}^1)) + H_{deco}(F_{L3}^1), \]
\[ F_{L1}^4 = H_{L1}^1(F_{L1}^2 + H_{deco}(F_{L2}^2)) + H_{deco}(F_{L3}^2), \]
\[ F_{L1} = H_{L1}(F_{SF}) = H_{L1}^S(F_{L1}), \]

where \( F_{L1}^j (i = 1, 2; j = 1, 2, 3, 4, 5) \) represents the output of each RRCAB in L1 and L2 level respectively, and \( H_{L1}^n (n = 1, 2, 3, 4, 5) \) denotes the RRCAB function. \( H_{deco} \) denotes the transposed convolution (with stride 2) function. \( F_{L1} \) and \( H_{L1} \) represent the network output and function of the L1 level respectively.

**L2 level** is composed of a downsample block and an upsampling block with a scale of 2 and several RRCABs. Different from the L1 level, we first use a 3 × 3 strided convolutional layer (with stride 2) followed by a LeakyReLU layer and a 3 × 3 convolution to conduct downsampling for input shallow features \( F_{SF} \). Several RRCABs are then used to extract and fuse the features from L3 level. Finally, a 3 × 3 convolution and a sub-pixel [15] convolution are applied to upsample the extracted features to the size of the HR image. The L2 level can be described as

\[ F_{d}^2 = H_d(F_{SF}), \]
\[ F_{d}^2 = H_{L2}^1(F_{L2}^1 + H_{deco}(F_{L3}^1)) + H_{deco}(F_{L3}^3), \]
\[ F_{L2}^4 = H_{L2}^1(F_{L2}^2 + H_{deco}(F_{L3}^2)) + H_{deco}(F_{L3}^4), \]
\[ F_{L2} = H_{up}(H_{L2}(F_{d})) = H_{up}(H_{L2}^S(F_{L2}^1)), \]

where \( H_d \) and \( F_d^2 \) denote the downsample function and the downsampled features respectively. \( F_{L2}^j (i = 1, 2, 3, 4, 5) \) and \( F_{L3}^j (j = 2, 4, 7, 9) \) denote the output of each RRCAB in L2 level and L3 level respectively. \( H_{L2}^n (n = 1, 2, 3, 4, 5) \) denotes the function of each RRCAB in L2 level. \( H_{up} \) is the upsampling function. \( F_{L2} \) and \( H_{L2} \) represent the network output and function of the L2 level respectively.

**L3 level** mainly refers to the structure of progressive up-sampling [19] to extract the informative features that are downsampled with a scale of 4. First of all, the extracted shallow features are operated by two downsampling blocks with a scale of 2. Then, the features are progressively fed into some RRCABs and upsampling blocks. In addition, unlike L1 and L2, we have introduced additional shortcut connection for better local feature fusion (LFF). The L3 level can be formulated as

\[ F_{L3} = H_{L3}^1(H_{L3}^{-1}(...H_{L3}(H_{d2}(H_{d1}(F_{SF}))))...)), \]
\[ F_{L3} = H_{L3}^1(H_{LFF}(H_{d2}(H_{d1}(F_{SF})), F_{L2}^3, F_{L3}^3)), \]
\[ F_{L3} = H_{L3}^1(H_{LFF}(H_{up1}(F_{L3}^2)))...), \]
\[ F_{L3} = H_{up2}(H_{LFF}(F_{L3}^5, F_{L3}^7, F_{L3}^9)), \]
\[ F_{L3} = F_{L3}^5 \]

where \( H_{L3}^j (i = 1, ..., 10) \) represents the function of each RRCAB in L3, and \( F_{L3}^j (j = 1, ..., 9) \) represents the output of each RRCAB. \( H_{d1}, H_{d2}, H_{up1} \) and \( H_{up2} \) are the downsampling and upsampling functions respectively. \( H_{LFF} \) represents the local feature fusion operation, which is composed of concatenation operation and 1 × 1 convolution. \( F_{L3} \) denotes the output of the L3 level.

Finally, the features extracted from L1, L2 and L3 levels are element-wise summed, and then sent to four RRCABs for further feature extraction. This procedure can be denoted as

\[ F_{DF} = H_{DF}(F_{L1} + F_{L2} + F_{L3}), \]

where \( H_{DF} \) is the deep feature extraction function composed of several RRCABs and \( F_{DF} \) is the extracted features.

### 2.3. Lightweight Recurrent RACB

Through some experiments, it can be seen that although RCAN can achieve comparable performance, it also brings a lot of parameters and memory overhead, which is not suitable for lightweight applications.

As shown in Fig. 2, we design a lightweight recurrent residual channel attention block (RRCAB). In our work, grouping convolution (with group 6) is applied to replace the 3 × 3 convolution in RACB, and a 1 × 1 convolution is added to fuse the features extracted by grouping convolution. In addition, we have added a recurrent structure to further improve the performance while remain the number of parameters unchanged. To better balance the model complexity and the quality of the generated image, each of RRCAB only looped 1 time. Based on RRCAB, we also design a more lightweight and effective SR module, named RRCAB-S, by removing the loop connections and reducing the number of the channels. The RRCAB can be described as

\[ F_{ca} = H_{ca}(F_{in}) = F_{in} \times H_{sig}(H_{1x1}(\sigma(H_{1x1}(H_{ga}(F_{in}))))), \]
\[ F_m = H_{rec}(F_{m-1} + H_{ca}(H_{1x1}(\sigma(H_{F_{get1}(\sigma(H_{get2}(F_{m-1}))}))))), \]

where \( F_{ca} \) and \( H_{ca} \) represent the output and function of the channel attention, respectively. \( H_{sig} \) is the sigmoid function, and \( H_{ga} \) denotes the global average pooling operation. \( H_{1x1} \)
is 1 × 1 convolution, and the superscript is the index of the layers. \( H_{gc1} \) and \( H_{gc2} \) represent grouping convolution at different layers. \( H_{rec} \) and \( \sigma \) denote recurrent function and ReLU, respectively. \( F_{m−1} \) and \( F_m \) represent the input and output of the module, respectively.

### 3. EXPERIMENTS

#### 3.1. Datasets and Metrics

As with the previous works \cite{5,7,11,17}, we also train our model with the DIV2K dataset \cite{18}, which has 800 high-quality RGB training images. For testing, we also use several widely used benchmarks: Set5 \cite{19}, Set14 \cite{20}, BSDS100 \cite{21} and Urban100 datasets. It also demonstrates the effectiveness of our basic RRCAB. We mainly focus on the effectiveness of our basic RRCAB. We mainly focus on the effect of the channel attention (CA), the effect of the feature fusion (FF, 1 × 1 convolution followed the second group convolution, as shown in Fig. 2) and the channel shuffle (CS) after group convolution. FF and CS are mainly used to fuse and shuffle the features extracted by the group convolution to enhance the representation capability of extracted features. It can be seen from Table 2 that the CA mechanism can improve the performance with little parameters increase. The FF mechanism can improve the performance with only a few additional parameters. This indicates that our proposed RRCAB can better balance the number of parameters and the performance in a way.

#### 3.2. Implementation Details

We attain the LR images by downsampling the HR images using the bicubic interpolation. To augment the training dataset, we randomly rotate and flip the 800 training images from DIV2K. In our work, the mini-batch size is set as 16, and the LR patches with a size of 48 × 48 are randomly cropped from the LR images as the input. We train our model with ADAM optimizer by setting \( \beta_1 = 0.9, \beta_2 = 0.999 \), and \( \varepsilon = 10^{−8} \). The learning rate is initialized to \( 1 \times 10^{−4} \), and decreased to \( 6.25 \times 10^{−6} \) with a cosine annealing. The proposed SR network is implemented under the PyTorch framework.

#### 3.3. Ablation Study

**Comparisons of different feature interaction schemes.** On the premise that the number of channels is 20 and the size of the input LR patches is 16 × 16, we have carried out the following experiments, including interactive connection (IC, as shown in Fig. 1), complex interactive connection (CIC, add an interactive connection from Level 3 to Level 1 on the basis of IC) and interactive connection without parameter sharing (NS, deconvolution in an interactive connection does not share parameters). From Table 1, we can have the following observations. First, the use of the feature interactions can improve SR performance by adding only a few parameters. Second, the CIC operation leads to an increase in the number of parameters but with little performance improvement.

**Comparisons of different basic modules in RRCAB.** In this part, we mainly perform some experiments to evaluate the effectiveness of our basic RRCAB. We mainly focus on the effect of the channel attention (CA), the effect of the feature fusion (FF, 1 × 1 convolution followed the second group convolution, as shown in Fig. 2) and the channel shuffle (CS) after group convolution. FF and CS are mainly used to fuse and shuffle the features extracted by the group convolution to enhance the representation capability of extracted features. It can be seen from Table 2 that the CA mechanism can improve the performance with little parameters increase. The FF mechanism can improve the performance with only a few additional parameters. This indicates that our proposed RRCAB can better balance the number of parameters and the performance in a way.

#### 3.4. Comparison with state-of-the-arts

To confirm the effectiveness of the proposed SR model, we compare our devised MSFIN with some state-of-the-art methods, such as SRCNN \cite{1}, VDSR \cite{2}, MemNet \cite{3}, DRCN \cite{8}, DRRN \cite{9}, LapSRN \cite{4}, IDN \cite{11}, CARN \cite{10}, MADNet \cite{13}, s-LWSR \cite{14} and IMDN \cite{12}. In our work, in order to better assess the performance of MSFIN-S and MSFIN, we divided these methods into two groups according to the number of parameters less than 600K and greater than 600K. We mainly evaluate these methods on SR tasks with a scale of 4. As depicted in Table 3, our model achieves comparable performance with acceptable number of parameters. In addition, Fig. 3 shows the visual results of each SR model on the Set14 and Urban100 datasets. It also demonstrates the effectiveness of our SR model.

In order to enhance the performance of our MSFIN, referring to RCAN, we also adopted a self-ensemble enhancement mechanism to our method, denoted as MSFIN-S+ and MSFIN+. It can be seen from Table 3 that our MSFIN-S+ and
### Table 3
Quantitative comparisons of MSFIN and other state-of-the-arts for scale factor 4 on Set5, Set14, BSD100 and Urban100 datasets. Best and second best results are highlighted and underlined.

| Algorithm          | Scale | Params(K) | Set5 PSNR | Set5 SSIM | Set14 PSNR | Set14 SSIM | BSDS100 PSNR | BSDS100 SSIM | Urban100 PSNR | Urban100 SSIM |
|--------------------|-------|-----------|-----------|-----------|------------|------------|-------------|-------------|-------------|-------------|
| SRCNN [1]          | 4     | 57        | 30.48     | 0.8628    | 27.49      | 0.7503     | 26.90       | 0.7101      | 24.52       | 0.7221      |
| DRRN [9]           | 4     | 297       | 31.68     | 0.8888    | 28.21      | 0.7720     | 27.38       | 0.7284      | 25.44       | 0.7638      |
| IDN [11]           | 4     | 553       | 31.82     | 0.8903    | 28.25      | 0.7730     | 27.41       | 0.7297      | 25.41       | 0.7632      |
| CARN-M [10]        | 4     | 412       | 31.92     | 0.8903    | 28.42      | 0.7762     | 27.44       | 0.7304      | 25.62       | 0.7694      |
| s-LWSR32 [14]      | 4     | 571       | 32.04     | 0.8930    | 28.15      | 0.7760     | 27.52       | 0.7340      | 25.87       | 0.7790      |
| MSFIN-S (Ours)     | 4     | 352       | 32.08     | 0.8934    | 28.46      | 0.7782     | 27.49       | 0.7332      | 25.85       | 0.7776      |
| MSFIN-S+ (Ours)    | 4     | 352       | 32.18     | 0.8945    | 28.55      | 0.7798     | 27.54       | 0.7344      | 25.97       | 0.7804      |
| VDSR [2]           | 4     | 665       | 31.35     | 0.8838    | 28.01      | 0.7674     | 27.29       | 0.7251      | 25.18       | 0.7524      |
| DRCN [8]           | 4     | 1774      | 31.53     | 0.8854    | 28.02      | 0.7670     | 27.23       | 0.7233      | 25.14       | 0.7510      |
| LapSRN [4]         | 4     | 813       | 31.54     | 0.8850    | 28.19      | 0.7720     | 27.32       | 0.7280      | 25.21       | 0.7560      |
| MemNet [5]         | 4     | 677       | 31.74     | 0.8893    | 28.26      | 0.7723     | 27.40       | 0.7281      | 25.50       | 0.7630      |
| CARN [10]          | 4     | 1592      | 32.13     | 0.8937    | 28.60      | 0.7806     | 27.58       | 0.7349      | 26.07       | 0.7837      |
| s-LWSR64 [14]      | 4     | 2277      | 32.28     | 0.8960    | 28.34      | 0.7800     | 27.61       | 0.7380      | 26.19       | 0.7910      |
| MADNet [13]        | 4     | 1002      | 31.95     | 0.8917    | 28.44      | 0.7780     | 27.47       | 0.7327      | 25.76       | 0.7746      |
| IMDN [12]          | 4     | 715       | 32.21     | 0.8948    | 28.58      | 0.7811     | 27.56       | 0.7353      | 26.04       | 0.7838      |
| MSFIN (Ours)       | 4     | 682       | 32.28     | 0.8957    | 28.57      | 0.7813     | 27.56       | 0.7358      | 26.13       | 0.7865      |
| MSFIN+ (Ours)      | 4     | 682       | 32.39     | 0.8971    | 28.66      | 0.7829     | 27.61       | 0.7370      | 26.25       | 0.7892      |

**Fig. 3.** Visual results of MSFIN with other SR methods on Set14 and Urban100 datasets.
MSFIN+ can outperform almost all other methods within the same parameter range.

4. CONCLUSIONS

In this work, we devised an effective multi-scale feature interaction network (MSFIN) for lightweight single image super-resolution. In particular, we constructed a multi-scale feature interaction module (MSFIM) mainly composed of lightweight recurrent residual channel attention blocks (RRCAB), which achieved comparable performance with fewer parameters and contributed to the generation of high-quality images. In addition, MSFIM and RRCAB can expand the receptive field of the network to take full advantage of the LR image features from various scales and interactive connection. Quantitative and qualitative experiments have proved that our proposed MSFIN can have a better trade-off between performance and the model complexity.

5. REFERENCES

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

[2] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in CVPR, 2016, pp. 1646–1654.

[3] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu, “Memnet: A persistent memory network for image restoration,” in ICCV, 2017, pp. 4539–4547.

[4] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, “Deep laplacian pyramid networks for fast and accurate super-resolution,” in CVPR, 2017, pp. 624–632.

[5] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang, “Multi-scale residual network for image super-resolution,” in ECCV, 2018, pp. 517–532.

[6] Jie Hu, Li Shen, and Gang Sun, “Squeeze-and-excitation networks,” in CVPR, 2018, pp. 7132–7141.

[7] Yulun Zhang, Kumpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu, “Image super-resolution using very deep residual channel attention networks,” in ECCV, 2018, pp. 286–301.

[8] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, “Deeply-recursive convolutional network for image super-resolution,” in CVPR, 2016, pp. 1637–1645.

[9] Ying Tai, Jian Yang, and Xiaoming Liu, “Image super-resolution via deep recursive residual network,” in CVPR, 2017, pp. 3147–3155.

[10] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” in ECCV, 2018, pp. 252–268.

[11] Zheng Hui, Xiumei Wang, and Xinbo Gao, “Fast and accurate single image super-resolution via information distillation network,” in CVPR, 2018, pp. 723–731.

[12] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang, “Lightweight image super-resolution with information multi-distillation network,” in ACM MM, 2019, pp. 2024–2032.

[13] Rushi Lan, Long Sun, Zhenbing Liu, Huimin Lu, Cheng Pang, and Xiaonan Luo, “Madnet: A fast and lightweight network for single-image super resolution,” IEEE TCYB, 2020.

[14] Biao Li, Bo Wang, Jiabin Liu, Zhiqian Qi, and Yong Shi, “s-lwsr: Super lightweight super-resolution network,” IEEE TIP, vol. 29, pp. 8368–8380, Aug. 2020.

[15] Wenzhe Shi, Shi Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in CVPR, 2016, pp. 1874–1883.

[16] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan, “Closed-loop matters: Dual regression networks for single image super-resolution,” in CVPR, 2020, pp. 5407–5416.

[17] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in CVPR workshops, 2017, pp. 136–144.

[18] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang, “Ntire 2017 challenge on single image super-resolution: Methods and results,” in CVPR workshops, 2017, pp. 114–125.

[19] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” in BMVC, 2012, pp. 1–10.

[20] Roman Zeyde, Michael Elad, and Matan Protter, “On single image scale-up using sparse-representations,” in ICCS. 2010, pp. 711–730, Springer.

[21] David Martin, Charless Fowlkes, Doron Tai, and Jitendra Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in ICCV. IEEE, 2001, vol. 2, pp. 416–423.

[22] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, “Single image super-resolution from transformed self-exemplars,” in CVPR, 2015, pp. 5197–5206.

[23] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE TIP, vol. 13, no. 4, pp. 600–612, Apr. 2004.