CNN-based bit-depth enhancement by the suppression of false contour and color distortion

Changmeng Peng*, Luting Cai†, Zhizhong Fu‡, Xiaofeng Li §.
University of Electronic Science and Technology of China, Chengdu, China
* 201511010304@std.uestc.edu.cn † tsaiting@std.uestc.edu.cn
‡ fuzz@uestc.edu.cn § xfli@uestc.edu.cn

***Abstract*—Although 10-bit monitors are getting popular, most of the available media sources are 8-bit. The inconsistence between the low-bit-depth media sources and high-bit-depth monitors should be properly solved to make full use of the high-bit-depth equipment. Simply converting low-bit-depth images/videos to high-bit-depth ones via zero-padding would result in false contour artifacts in smooth region, which greatly degrades the visual quality. In this paper, a novel auto-encoder like CNN model is proposed to convert low-bit-depth images to high-bit-depth ones. Our method can significantly suppress false contour by the use of vgg_loss (mean square error computed on pre-trained VGG-19 feature maps). However, significant color distortion would be found in some results if only vgg_loss is used. In order to suppress color distortion, $range\_loss$ is proposed which restrains the difference between the resultant pixel values and the zero-padded ones within the range of [0, S), where S is the re-quantization step. Benefit from the novel network model and the designed loss function consisting of $range\_loss$ and vgg_loss, the proposed method has comparable objective metric with state-of-the-art. In particular, our method achieves better visual quality by significantly suppressing false contour artifacts and color distortion. Those conclusions are proved by experiments, and our code can be found at https://github.com/pengcm/BE-AUTO-ext.**

Fig. 1. The example of false contour artifacts. The lower part is the original 16-bit image, and the upper part is uniformly quantified from 16-bit to 4-bit, then enhanced back to 16-bit via ZP.

I. Introduction

Good watching experience has always been the target of the audiences and multimedia industry. It leads to the transitions from black-white videos to RGB-color ones; from VGA (640x480) modes to HD(1280 x 720) modes, and the forthcoming 4K/8K UHD modes in spatial resolution range [1]; and from conventional 8-bit to 10-bit per pixel color channel in bit-depth range. Although the monitors are going to support 4k/8k and 10-bit display, most of the current media sources are of lower bit-depth and spatial resolution. Super-resolution is there to expand the spatial resolution. Corresponding to super-resolution in spatial resolution expansion, technology used to expand bit-depth is called bit-depth enhancement.

To display low-bit-depth images on high-bit-depth monitors, bit-depth should be expanded. The most basic and simple way is ZP (Zero-Padding), which adds zeros after the least significant bits to get the target bit depth. However, ZP results have severe false contour artifacts as depicted in Fig.1. The lower part of Fig.1 is the original 16-bit image, and the upper part is gotten by quantizing the original image to 4-bit, then expanding back to 16-bit via ZP. As is presented in Fig.1, false contour can be seen in smooth areas, which greatly degrades the visual quality.

With the obvious incompatibility between the popularizing 10-bit monitors and existing 8-bit media sources, researches on bit-depth enhancement for better visual quality are of great importance. A close but different issue is inverse tone mapping iTM [2], where high dynamic range (HDR) images with hallucinated details in local minimum/maximum regions are reconstructed from low dynamic range (LDR) images. Those details are lost because of the non-linear tone mapping, or the limited exposure range, which results in detail loss in over/under-exposure regions. However, bit-depth enhancement aims at recovering the lost detail caused by limited quantization levels. As iTM methods are used for different detail loss type, they are not suitable for bit-depth enhancement tasks.

Many bit-depth enhancement methods have been proposed in recent years. As simple as ZP, bit replicate BR [3] replicates MSB (most-significant-bits) to the newly extended LSB (least-significant-bits), but false contour artifacts are still severe. Filtering based methods use non-linear filters with adaptive window size, such as [4], can get better visual quality compared with ZP and BR. As is shown in [5] and ACDC [6], properly making use of image prior, such as smooth prior and/or sparsity characteristics of image surface, can produce visual results of better quality. Optimization based methods such as ACDC [6], MRC [7] and [8] model the bit-depth enhancement task as minimization problems. By formulating the relationship between the LBD (Low-Bit-Depth) and the generated HBD (High-Bit-Depth) with probability theory, con-

This work is supported by the Natural Science Foundation of China (61671126) and Science and Technology Plan Project of Sichuan Province (2019JDR0022).

tour artifacts are relieved by maximizing a posteriori. Content-adaptive means like [9]–[11] take content information into consideration when estimating the added bits' values. IPAD [12] novelly introduces intensity potential field to model the relationship between pixels and gets state-of-the-art performance.

Compared with the flourish research on deep learning-based methods for image super-resolution, only two CNN-based methods [1], [13] are proposed for bit-depth enhancement. In [1], Liu et al. introduce CNN to solve image bit-depth enhancement task. As is stated in [1], although MSE (Mean Square Error) loss computed on the input and output images is commonly used in image super-resolution, it can't suppress false contour artifacts when used in bit-depth enhancement tasks. Alternatively, they use perceptual loss (MSE on VGG-19 feature maps of the ground truth image and the generated one) as the minimization function. Meanwhile, convolution layers are also replaced by deconvolution to get more realistic results. Even though BE-RTCNN [1] relieves false contour artifacts, color distortion is obvious in some test images as depicted in Fig. 4(b). Ref [13] is an auto-encoder like CNN model for video bit-depth enhancement.

To better solve the task of bit-depth enhancement, a new CNN-based method is proposed in this paper. Our method is an auto-encoder like CNN model, where the encoder is realized with convolution layers to extract features, which are then used by the full-deconvolution implemented decoder. To make efficient data flow in the network and overcome the gradient-vanish problem in deep neural network, skip-connections are used to link the corresponding layers in the encoder and the decoder modules. To suppress false contour artifacts and color distortion in previous methods, we simultaneously use perceptual loss used in [1] and the loss named $range\_loss$ originally proposed in this paper. $Range\_loss$ is a piecewise punishment function, which constrains the difference between the pixel values of the resultant image and the input image within the range of quantization bin. Experiments show that the proposed method enhances bit-depth effectively with comparable objective metric with state-of-the-art. In particular, our method achieves better visual quality by significantly suppressing false contour artifacts and color distortion.

The rest of this paper is organized as follows. Section II presents the problem modeling and our network structure, perceptual loss (vgg_loss), and the $range\_loss$ we propose. Experiments are presented in section III with comparison between our method and other related algorithms both visually in figures and objectively on PSNR. Conclusions are presented in section IV.

II. PROPOSED ALGORITHM

A. Problem modeling

Quantizer is used to convert the continuous luminous intensity to integer value with finite binary bits. The number of bits is called bit-depth. Common bit-depth are 8-bit,10-bit and 16-bit. Images of different bit depths vary in the diversity of colors, and the more bits, the more color levels and realistic visual quality. Meanwhile, the more data to be transmitted and stored.

The quantization methods can be divided into uniform quantization and non-uniform quantization, where uniform quantization is the most commonly used. Under uniform quantization, converting a high-bit-depth image into low-bit-depth one is just abandoning the n least-significant-bits, where n is the gap of bit-depth. Bit-depth enhancement is just the reverse procedure, which converts low-bit-depth image to high-bit-depth one. Zero-Padding ZP is the simplest way. ZP extends the least-significant-bits of the low-bit-depth image naively with zeros to get the wanted bit-depth. Although ZP is simple, severe false contour artifacts are encountered in the results. As is shown in Fig. 1, contour artifacts appear in flat gradient areas under ZP, which greatly degrade the visual quality. The main purpose of bit-depth enhancement is to find a function mapping the LBD image to HBD, and try to maximize the likelihood between the real HBD and the reconstructed one with the help of some image prior. This can be formally expressed by (1), where l measures the similarity between the ground truth HBD and the reconstructed one; f is the function mapping LBD image to HBD; p is prior term like sparsity [5], content [10], [11], or context [9].

$$\arg\max_{f} \; l(\boldsymbol{I^{HBD}}, f(\boldsymbol{I^{LBD}})) \quad s.t. \quad p(\boldsymbol{I^{LBD}}). \tag{1}$$

Deep learning is now the most hot research area. It is widely used in recommendation system, image classification, object tracing, image super-resolution, etc. It is a data-driven methodology in the sense that it uses great amount of data to train the neural network. In the training process, loss function and gradient decent guide the network to digit the inner universal operation that non-linearly maps the input to the corresponding label. When using CNN to implement bit-depth enhancement, (1) can be instantiated as (2), where $\boldsymbol{\theta}$ is the weights of the network.

$$\arg\min_{\boldsymbol{\theta}} \; loss(\boldsymbol{I^{HBD}}, CNN_{\boldsymbol{\theta}}(\boldsymbol{I^{LBD}})) \tag{2}$$

B. Our CNN-model

Our network is an auto-encoder like one consists of encoder module and decoder module as depicted in Fig. 2. In the encoder module, we use 5 convolution layers with incremental filter number to extract features at different levels starting from the input image. In the decoder module, 5 deconvolution layers are used to reconstruct image from the features extracted by encoder module. To overcome range shift of deconvolution layers and relieve gradient vanishing problem, we add a batch-normalization-layer [14] after each deconvolution layer. Skip-connections link the convolution layers in encoder and the corresponding deconvolution layers in decoder to make better data flow and help the gradient decent. Activation layers used in our network are all ReLU as [1] has done.

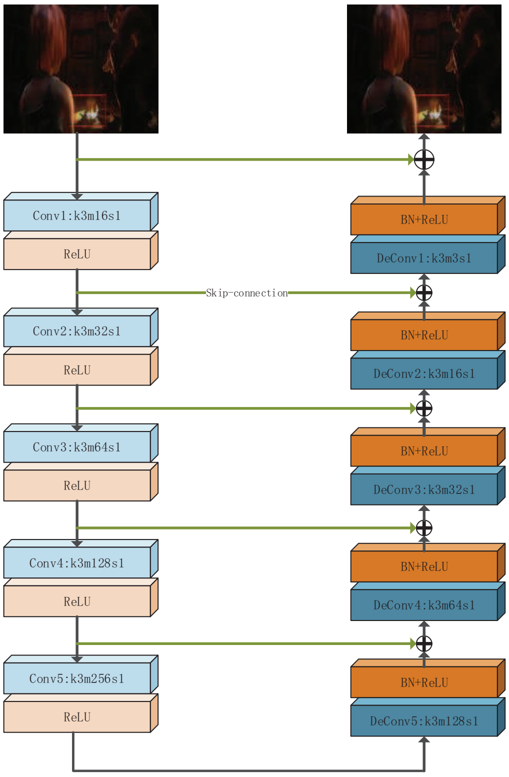


Fig. 2. Network model of our algorithm. Our network takes RGB images as input and generates RGB images; $k{*}m{*}s{*}$ is the parameters of corresponding conv/deconv layer where k means kernel size ($k \times k$), m means the number of feature maps, s means conv/deconv stride. Green lines are skip-connections, and the skip-connected feature maps are pixel-wise added.

C. Loss function

Loss function is critical to the network, for it greatly decides the robustness and results' quality of the network. Commonly used loss function in related super-resolution tasks is MSE loss, and it is a plain mean squared difference of pixel values between the generated image and the ground truth. As is stated in [1], the exclusive use of MSE loss in bit-depth enhancement can not suppress false contour artifacts. In SRGAN [15], perceptual loss is used to yield super-resolutioned images with better realistic and natural textual details. And authors of SRGAN show that perceptual loss implemented as MSE on VGG-features, i.e. vgg_loss as we formulate in (3), can better realize that target than plain MSE on the image pair, i.e. mse_loss stated by (4). Inspired by this, we use MSE on VGG-features as a component of our loss function and we call this component vgg_loss, which is formally expressed by (3). It should be mentioned that vgg_loss is used as the unique loss function in [1], [16] where it is called perceptual loss. As is depicted in Fig. 4(b), exclusively using vgg_loss can truly suppress false contour artifacts at some level, but color distortion can be obviously seen in some cases. It can be found that pixel values in color-distortion area exceed the valid range seriously. For example, when we extend a uniform-quantified 4-bit image back to 16-bit, the valid range for pixel values of the result is [*lsb* + 0, *lsb* + 2e(16-4)), where *lsb* is the pixel value of the considering pixel of the zero-padded image. Based on this prior that pixel value must lie in the valid range, we propose another loss function component called $range\_loss$, as is stated by (5). It can be seen as a piecewise punishment function. To suppress false contour artifacts and color distortion at the same time, we use a two-component loss function which consists of vgg_loss and $range\_loss$ formulated by (6), where $\lambda 1$ and $\lambda 2$ are the weights.

TABLE I
SYMBOL TABLE

symbol	meaning
I, i	the number and index of the layer of G
J_i, j	the number and index of the i-th layer's channels
W, H	the width and height of the image/feature
x, y	the coordinate of pixels
G	the CNN which is used to generate the results
I^{HBD}	the original HBD
$\hat{I}^{HBD}$	the generated HBD
S	the length of valid range
$\omega 1, \omega 2$	the punishment factor for within/out of the valid range

$$vgg\_loss = \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}} \frac{1}{IJ_iW_{i,j}H_{i,j}}(G(\boldsymbol{I^{HBD}})_{x,y}^{i,j} - G(\boldsymbol{\hat{I}^{HBD}})_{x,y}^{i,j})^2 \tag{3}$$

$$mse\_loss = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}(\boldsymbol{I_{x,y}^{HBD}} - \boldsymbol{\hat{I}_{x,y}^{HBD}})^2 \tag{4}$$

$$range\_loss = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H} P(\boldsymbol{\hat{I}^{HBD}}, \boldsymbol{I^{LBD}})_{x,y} \quad \text{where}$$
$$P(\boldsymbol{A}, \boldsymbol{B})_{x,y} = \begin{cases} \omega 1 & \boldsymbol{A}_{x,y} \in [\boldsymbol{B}_{x,y}, \boldsymbol{B}_{x,y} + S) \\ \omega 2 & \boldsymbol{A}_{x,y} \notin [\boldsymbol{B}_{x,y}, \boldsymbol{B}_{x,y} + S) \end{cases} \tag{5}$$

$$loss = \lambda 1 \cdot vgg\_loss \; + \; \lambda 2 \cdot range\_loss. \tag{6}$$

III. EXPERIMENTS

In this section, we will valid the effectiveness of our network structure and loss function form. After that, comparison with most representative methods for bit-depth enhancement are presented, both in objective metric PSNR and in subjective visual evaluation.

A. Experiment settings

As our algorithm is CNN-based, and [1] is the only method using CNN for image bit-depth enhancement, we use the same dataset as [1] for comparison. The train set contains 1000 frames randomly selected from the 20-thousand Sintel dataset [17], which is 16-bit 436 x 1024. The test set is also the eight frames used as test set in [16], which is the conference version of [1], and the names of the eight frames are strictly the same as what they are in [16]. The train is performed on a NVIDIA 1050Ti GPU with 3GB memory. Batch size is set to 8 with 120 epochs. For every image, we randomly select a 96 x 96 patch in every epoch. Learning rate is 1e-4. Adam optimizer with beta1=0.9 is taken. $\lambda 1$ and $\lambda 2$ are set to 0.6 and 5e-6, $\omega 1 = 0$, $\omega 2 = 65535$. We linearly compress the 16-bit images to 4-bit ones, then use Zero-Padding to pad them back to 16-bit, which is used as the network input. All the following results are conducted in this 4-bit-to-16-bit scenario. In order to avoid local minima in early training step, we use mse_loss for the first epoch to pre-train the network.

B. Convolution vs. Deconvolution

The difference between convolution and deconvolution mainly lies in the logical mode. Convolution is bottom-up, which extracts features from lower-abstract-level feature maps, e.g. from raw image to features like edges, lines; On the contrary, deconvolution tries to reconstruct lower-level features top-down [18]. Following this intuitional guideline, we use convolution layers to extract features and reconstruct image from them using the following deconvolution layers. This methodology is just validated by Fig. 3, where full-conv/deconv means all the layers in the network are implemented as convolution/deconvolution layers, and conv-deconv represents the methodology we take, i.e. using convolution layers to construct encoder module and deconvolution for decoder module. As we can see from Fig. 3, network implemented as conv-deconv version dominates full-conv on all the 8 test images which are randomly selected from *Sintel* dataset as stated in [16]; full-deconv implementation performs better than conv-deconv on IMG2, but it is worse than conv-deconv on the others. In addition, it obviously lacks of robustness.

C. Color distortion and range-loss

The exclusive use of vgg_loss can suppress false contour artifacts, but color distortion appears in some results, which is also reported in [1]. Color distortion degrades visual quality as false contour artifacts do, so we need to suppress it as well for better quality. As we have explained in section II, pixel values in color distorted regions seriously exceed the valid range. Based on this founding, we propose $range\_loss$. The effectiveness of $range\_loss$ is shown by Fig. 4. We can see that color distortion happens at the flame and the right shoulder of the girl from Fig. 4(b), which is the result without the use of $range\_loss$. Color distortion can be significantly suppressed by $range\_loss$ as depicted by Fig. 4(c).

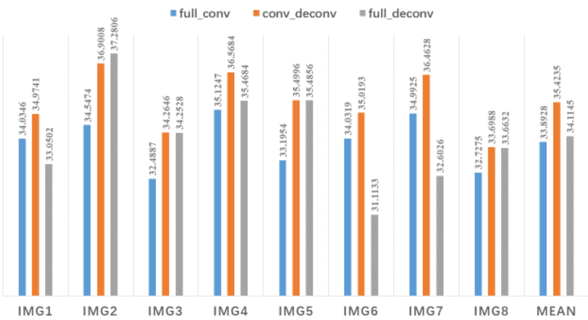


Fig. 3. Conv VS. Deconv. The comparison of the effect conv/deconv on the performance of the network evaluated on PSNR. The last three bars are gotten via the plain average of the corresponding eight images.

(a) ground-truth

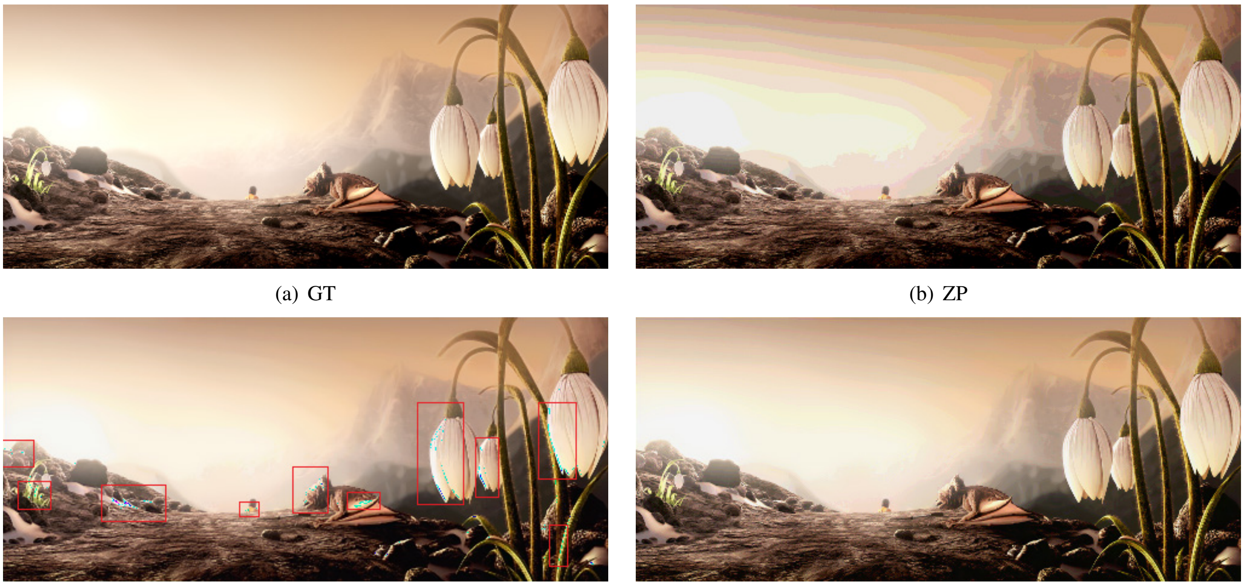
(b) without $range\_loss$

(c) with $range\_loss$

Fig. 4. Color distortion suppression with $range\_loss$. (a) is the ground truth image;(b) is the result of exclusively using vgg_loss without $range\_loss$;and (c) is the result of using vgg_loss with $range\_loss$ simultaneously.

D. Comparison with state-of-the-art

In this section, our method is compared with state-of-the-art bit-depth enhancement methods, including ZP, MIG, BR [3], MRC [7], ACDC [6], CRR [11], CA [10], IPAD [12] and BE-RTCNN [1]. Table II lists the PSNRs of the eight test



(a) GT (b) ZP

(c) BE-RTCNN (d) ours

Fig. 5. Visual comparison with state-of-the-art.

(a) GT (b) ZP

(c) BE-RTCNN (d) ours

Fig. 6. Visual comparison with state-of-the-art.

images under the listed methods. In that table, PSNR values of BE-RTCNN are extracted from [16], which is the conference paper of [1]. Values in bold mean the best, and the second-best is emphasized by underline. As we can see in Table II, our method can produce results which have higher PSNR than BE-RTCNN(the only one CNN-based method for image bit-depth enhancement), like IMG2 and IMG4, even though not as good as BE-RTCNN in the other cases of the test set. Our method, on average, has second-best objective metric PSNR when compared with state-of-the-art.

The core advantage of our method is producing results of better visual quality, and this is depicted by Fig. 5 and Fig. 6. In those two figures, Fig. 5(a) and Fig. 6(a) are the original images; Fig. 5(b) and Fig. 6(b) are the results of ZP, where false contour artifacts are obvious, and they degrade visual quality severely. Fig. 5(c) and Fig. 6(c) are the BE-RTCNN results. BE-RTCNN suppresses contour artifacts at some level, but color distortion happens at the border of the flowers, the wing of the dragon and on the roofs, etc. Color distortion is emphasized by using red rectangular boxes in those two sub-figures. Fig. 5(d) and Fig. 6(d) are the results of our method. Compared with ZP in Fig. 5(b) and Fig. 6(b), our

TABLE II
COMPARISON WITH STATE-OF-THE-ART ON PSNR

	ZP	MIG	BR	MRC [7]	ACDC [6]	CRR	CA	IPAD [12]	BE-RTCNN[1][1]	ours
IMG1	29.9452	30.7667	30.0000	32.2148	33.9726	29.7819	35.0805	34.4545	**35.1052**	34.7972
IMG2	28.8946	32.2364	28.9457	33.0027	35.7154	35.7981	36.2761	36.4827	36.4582	**36.8795**
IMG3	28.5461	31.7070	28.5954	33.7119	32.6138	32.7263	34.9535	34.6487	**35.2115**	34.1946
IMG4	29.3589	30.0110	29.4127	30.9309	34.9019	31.5720	35.2872	36.4388	35.9715	**36.8685**
IMG5	28.8932	30.8676	28.9440	31.6325	34.1867	34.6148	**36.1632**	35.7061	35.9674	35.1526
IMG6	31.8807	33.0015	31.9376	35.5890	30.0466	27.0007	32.8041	31.1019	**37.9192**	33.6237
IMG7	31.4774	32.7361	31.5399	34.8553	32.3445	28.0670	34.5621	33.2172	**37.3571**	35.3399
IMG8	28.7163	31.7551	28.7653	32.8013	32.9975	31.1205	**34.8089**	33.8435	34.7358	34.2022
MEAN	29.7141	31.6352	29.7676	33.0923	33.3474	31.3352	34.9920	34.4867	**36.0907**	35.1323

[1] means the results we directly extracted from [16]

method suppress false contour artifacts in smooth gradient regions significantly. What's more, our method can suppress color distortion which arises in BE-RTCNN's results.

Proven by Fig .5, Fig. 6 and Table II, our method can enhance image bit-depth effectively. Objectively, our method has second-best PSNRs among currently known methods. And for visual result, which is the core goal of bit-depth enhancement, our method produces better visual quality by significantly suppressing false contour artifacts and color distortion. It should be mentioned that, the better visual results of our method comes from the novel auto-encoder like network structure and the two-component loss function we propose.

E. Logical analysis

Loss function proposed in this paper consists of vgg_loss and $range\_loss$. These two terms are used for the suppression of false contour and color distortion respectively. For false contour suppression, we use vgg_loss rather than mse_loss. mse_loss pursues pixel-wise likelihood and treats pixels separately, while vgg_loss is based on the features extracted by the convolution layers. vgg_loss aims at minimizing the difference of context between the generated image and the label at multi abstract levels (convolution layers of VGG-19), so it can produce better visual quality and photo-realistic texture in result images than mse_loss.

The color of a pixel is decided by the values of all the color-channels (such as RGB) jointly. Once a color component changed, color distortion may arise, and distortion degree is in proportion to the gap between the result and the ground-truth. Pixel values in color distortion areas are different from ground-truth and exceed valid range seriously. Based on the range prior, $range\_loss$ restricts the guessed values within valid range via punishment, thus can lessen the difference between the guessed pixel value and the ground truth, and suppress color distortion.

IV. CONCLUSIONS

In this paper, we propose an auto-encoder like CNN model, where the encoder and decoder are implemented with 5 convolution layers and 5 deconvolution layers respectively. Skip connections are also used to make the training easier. For better visual results, we use vgg_loss instead of mse_loss mostly used in feed-forward networks. The exclusive use of vgg_loss can suppress false contour artifacts, but color distortion arises. To deal with this, we introduce $range\_loss$ based on the finding that pixel values in color distorted region exceed valid range. With the novel auto-encoder like network model and the loss function consisting of $range\_loss$ and vgg_loss, our method produces results which are comparable with the state-of-the-art PSNRs. In particular, our method produces better visual quality by significantly suppressing false contour artifacts and color distortion as is shown by Fig. 5 and Fig. 6.

REFERENCES

[1] Y. Su, W. Sun, J. Liu, G. Zhai, and P. Jing, "Photo-realistic image bit-depth enhancement via residual transposed convolutional neural network," *Neurocomputing*, vol. 347, pp. 200 – 211, 2019.

[2] Y. Endo, Y. Kanamori, and J. Mitani, "Deep reverse tone mapping," *ACM Transactions on Graphics (Proc. of SIGGRAPH ASIA 2017)*, vol. 36, Nov. 2017.

[3] R. Ulichney and S. Cheung, "Pixel bit-depth increase by bit replication," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 3300, 05 1998.

[4] A. Taguchi and J. Nishiyama, "Bit-length expansion by inverse quantization process," pp. 1543–1547, 01 2012.

[5] P. Wan, O. Au, K. Tang, and Y. Guo, "Image de-quantization via spatially varying sparsity prior," pp. 953–956, 09 2012.

[6] P. Wan, G. Cheung, D. Florencio, C. Zhang, and O. Au, "Image bit-depth enhancement via maximum-a-posteriori estimation of graph ac component," *2014 IEEE International Conference on Image Processing, ICIP 2014*, vol. 25, pp. 4052–4056, 01 2015.

[7] G. Mittal, V. Jakhetiya, S. Jaiswal, O. Au, A. Tiwari, and W. Dei, "Bit-depth expansion using minimum risk based classification," 11 2012.

[8] J. Woo, S. Lee, and W. Choe, "Bit depth expansion via estimation of bit value expectation," *Electronic Imaging*, vol. 2016, pp. 1–6, 02 2016.

[9] J. Liu, W. Sun, P. Jing, J. Yu, and Y. Su, "Improving bit-depth expansion via context-aware mmse optimization (camo)," *IEEE Access*, vol. PP, pp. 1–1, 07 2018.

[10] P. Wan, O. Au, K. Tang, Y. Guo, and L. Fang, "From 2d extrapolation to 1d interpolation: Content adaptive image bit-depth expansion," pp. 170–175, 07 2012.

[11] C. Hong Cheng, O. Au, C. H. Liu, and K. Yue Yip, "Bit-depth expansion by contour region reconstruction.," pp. 944–947, 05 2009.

[12] J. Liu, G. Zhai, X. Yang, and C.-W. Chen, "Ipad: Intensity potential for adaptive de-quantization," *IEEE Transactions on Image Processing*, vol. PP, pp. 1–1, 02 2018.

[13] J. Liu, P. Liu, Y. Su, P. Jing, and X. Yang, "Spatiotemporal symmetric convolutional neural network for video bit-depth enhancement," *IEEE Transactions on Multimedia*, vol. PP, pp. 1–1, 02 2019.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456, JMLR.org, 2015.

[15] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 09 2016.

[16] J. Liu, W. Sun, and Y. Liu, "Bit-depth enhancement via convolutional neural network," in *Digital TV and Wireless Multimedia Communication* (G. Zhai, J. Zhou, and X. Yang, eds.), (Singapore), pp. 255–264, Springer Singapore, 2018.

[17] X.Foundation, "Xiph.org." https://www.xiph.org/., 2016. Accessed May 1,2019.

[18] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2528–2535, June 2010.