Saliency Map-aided Generative Adversarial Network for RAW to RGB Mapping

Yuzhi Zhao *[1], Lai-Man Po[1], Tiantian Zhang[1], Zongbang Liao[2], Xiang Shi[2], Yujia Zhang[1], Weifeng Ou[1], Pengfei Xian[1], Jingjing Xiong[1], Chang Zhou[1], and Wing Yin Yu[1]

[1]City University of Hong Kong, Hong Kong SAR, China
[2]Huazhong University of Science and Technology, China

Abstract

RAW files are widely applied in cameras and scanners as storage because they contain original optical data. Different cameras usually process the RAW files using diverse algorithms that are incompatible. To address the issue, we propose a general transformation method for cross-camera RAW to RGB mapping based on Generative Adversarial Network (GAN). Moreover, we propose a saliency map-aided data augmentation technique and the saliency maps are produced by Saliency GAN (SalGAN). Given RAW file as an input, it jointly predicts the RGB image and corresponding saliency map to enhance perceptual quality in the generated image. The proposed architecture is trained on the Zurich RAW2RGB (ZRR) dataset. Experimental results show that our method can generate more clear and visually plausible images than state-of-the-art networks.

1. Introduction

The RAW files contain the original data from optical sensors of either a digital camera, a phone camera, or an image scanner. Normally, RAW files have not been processed and therefore represent a wide dynamic range. However, RAW files cannot be directly printed as visible images, although they contain information to construct a human-readable format with high quality. To obtain the photographic rendering of the scene, traditional pipeline involves five operations: decoding the RAW files, interpolating the raw pixels into color pixels (demosaicing), white balancing and noise reduction, color translation and tone reproduction (e.g. sRGB color space), finally compressing the result (e.g. JPEG format). Different cameras often come with proprietary conversion techniques while it is also difficult to develop uniform standard for processing RAW format. In this paper, we investigate a general mapping for RAW files, trying to generate photorealistic RGB representations.

Since human retina is most sensitive to green light, Bayer [2] proposed a specific color composition to mimic the physiology of human perception. It is designed to maximize clarity of the perceived luminance. The RAW files captured by cameras follow this scheme and is called 'Bayer Pattern'. There are multiple types of storage for RAW files, all of which use two times green pixels compared with red or blue, hence are composed of RGGB, RGBG, BGGR, or GRGB. Basically, each pixel of RAW files is designed to record only one of the RGB channels, therefore it cannot specify real color of corresponding pixel in visible format. To obtain the RGB images, the 'demosaicing' process interpolates every neighboring four pixels in RAW format into three channels in RGB format. Take a simple 'demosaicing' approach for instance, a RAW pixel filtered by green Bayer filter provides precise value in output green channel. The red and blue channels for this output pixel are computed by adjacent RAW pixels. Specifically, two near red or blue RAW pixels can be used to predict the output red or blue channels for current location, respectively. Normally, traditional 'demosaicing' methods may produce discontinuous and false colors due to minor local perception field. Moreover, since different companies often adopt separate algorithms to visualize RAW files, it is challenging to transfer RAW files into visible format using discordant techniques. To address these problems, we utilize deep learning-based techniques that model long-range pixel relations and learn a general mapping for images captured with different cameras. As Generative Adversarial Network (GAN) [5] has achieved great fitting ability to transform the images from diverse domains, we define RAW files and RGB photos as two related domains and learn the general mapping from RAW to RGB.

Image domain transfer aims to learn the internal relationship and mapping among images within two or more domains. Recent years have witnessed a wide range of applications to enforce translation among different types of

*Corresponding author: yzzhao2-c@my.cityu.edu.hk

2473-9944/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCVW.2019.00428

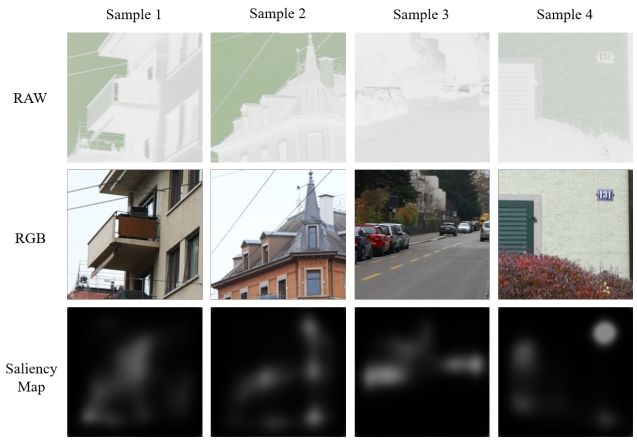

Figure 1. Some training samples. The first row and second row visualize the input RAW files and target RGB photos in ZRR training set. The last row shows the saliency maps generated from the pre-trained SalGAN. For each sample, the training images are paired. Please visit `https://github.com/zhaoyuzhi/RAW2RGB-GAN` to see more examples and to try our model and code.

source data, such as grayscale pixels [14], semantic maps [32], edges with texture [35], and class labels [19]. The data for domain transfer can be categorized into paired [14] and unpaired [37], while paired images mean that the pixels are all aligned. RAW to RGB is a kind of paired image domain transfer task. For each pair, the RAW input contains four channels (RGBG) while the output is RGB image. Basically, RAW to RGB is a new topic in domain transfer area and there are limited relevant researches. Therefore, we utilize the general algorithm Pix2Pix [14] as a baseline, which is a conditional GAN [22] system that models paired image transformation.

To enhance the perceptual vividness, we propose a novel saliency map-aided GAN framework and train it on the newly collected Zurich RAW2RGB (ZRR) dataset [11] in AIM 2019 RAW to RGB mapping challenge [11]. The dataset contains over 90000 aligned local image patches captured with Canon 5D Mark IV and Huawei P20 cameras. The visible JPEG format photos are taken from Canon camera. The RAW files are from Huawei camera, then reshaped into 4-channel PNG format for saving. Actually, the four channels represent the RGBG (transparency channel = G) that remain the value of original data. There are 89000 training pairs, 2139 validation pairs, and 2139 testing pairs in the dataset. The main challenge for learning the mapping lies in limited training data hence effective data augmentation methods are vital. Normal data augmentation approaches such as random cropping, rotation, and flipping have been widely used in image generation tasks for optimization. However, they perform only physical transformations without semantic information, which is significant for domain transfer. To better describe semantics, the saliency maps of output RGB images are involved in our model, which represent attention area of human. Saliency map is a set of contours extracted from the image and high response area indicates more attention. We use the pre-trained Saliency GAN (SalGAN) [26] to generate saliency maps corresponding to output RGB images and make them as proxy target for the system. We show some training samples in Figure 1. Apart from predicting RGB images, our model also produces saliency maps, then they are used to scale the pixel level loss as a special type of data augmentation.

Compared with the baseline, the main contributions of this paper are as follows:

1) We propose a novel saliency map-based data augmentation method to enhance the performance of limited training data;

2) We utilize an efficient U-ResNet generator architecture that generates 28 images per second on single GPU;

3) We experimentally demonstrate the proposed model produces high-quality RGB images.

2. Related work

For paired image-to-image translation tasks, there are many commonalities between RAW to RGB and colorization. Firstly, both targets are to produce perceptually plausible RGB images, while the inputs are RAW files (4 channels) and grayscale images (1 channel), respectively. Secondly, the input and output for either training or testing are paired, which means the pixels at same position are directly related. Thirdly, both tasks are kinds of image generation and share similar algorithms. However, there are also some main differences. Normally, colorization algorithms utilize a whole image to learn the mapping, while this paper tackles the problem that recovers the real scene from a certain patch of a whole RAW image. Additionally, recent state-of-the-art colorization approaches are trained on a very big dataset like ImageNet [29], while the ZRR dataset only contains image pairs less than one tenth of ImageNet. We firstly investigate colorization algorithms, then introduce some GAN frameworks for image enhancement. Finally, we conclude all the networks and analyze the design of proposed architecture.

The existing colorization algorithms can be briefly categorized into three classes: scribble-based [18], exemplar-based [34], and fully automatic [4, 12, 17, 36]. Given some color hints, scribble-based approaches propagate them to the rest of target grayscale image. It is similar to the painting process of human. In contrast, exemplar-based approaches extract the color information of a reference image, then apply it to the grayscale image by matching their spatial features. However, these methods require much additional information in the form of color scribble and refer-

ence image. An alternative approach is to train an automatic end-to-end colorization system. Recently, fully automatic colorization methods learn the general color distribution of natural images from a large dataset. Normally, these methods have better generalization ability to different types of images without any intervention.

Recently, Cheng et al. [4] proposed a pioneering deep learning-based colorization system, which directly colorizes image based on the handcrafted features. The features are three-level given by patch, DAISY, and semantic descriptors. However, the network structure is too limited to improve the performance. Instead of using handcrafted features, Larsson et al. [17] adopted an end-to-end network. Since colorization task relies much on semantic information, they utilized the hyper-column descriptors from a pre-trained VGG-16-Gray [30] network to predict the hue and chroma for each pixel. To reduce computing time, Iizuka et al. [12] proposed a multi-task colorization system. The mainstream is an auto-encoder, while there is an additional branch that classifies the input grayscale images. The classification branch enhances the whole system to extract the semantics of input, which improves the colorization quality. On the other hand, Zhang et al. [36] transferred colorization into a classification task. They separated all the color combinations into different discrete values and trained the system supervised by them.

Generative Adversarial Network (GAN) [5] has shown its advance in data generation in recently years. It allows the generated data to fit into the ground truth data by minimizing divergence of both distributions. For the GAN framework, generator G captures the data distribution, and discriminator D judges whether the sample is ground truth or produced from generator. During the training process of GAN, G and D are trained alternatively until G has enough ability to approximate the real data distribution. However, the adversarial training process is oscillating, which is easy to fall into model collapse. To stabilize the training of GAN, Radford et al. [27] proposed a novel architecture called Deep Convolutional Generative Adversarial Network (DCGAN). It has certain architectural constraints and hyperparameter settings therefore generator learns good representations of images for generative modeling. Moreover, regular GAN hypothesizes the discriminator as a binary classifier while minimizes Jensen-Shannon-divergence (JSD) between the generated distribution and ground truth at training. However, it usually leads to adversarial training failure due to gradient vanishing. To address this problem, diverse objective functions have been proposed. For example, the well-known LSGAN [21] adopt a least squares loss function for the discriminator that minimizes Pearson χ^2 divergence. The WGAN [1] empirically minimizes an efficient approximation of the Earth Mover distance, which is enhanced by WGAN-GP [6] by adding the gradient penalty.

There are some frameworks [9, 10, 14, 32] that combine conditional image generation and GAN for image enhancement. Ignatov et al. [9] proposed a novel enhancer that transforms the ordinary images into DSLR quality photographs with arbitrary sizes. To effectively train the model, they collected a large dataset including smartphone photos as input while DSLR camera photos as output. However, this model required aligned training pairs. It was improved by adding an additional adversarial color loss [10] that makes whole system ignore the trivial location disparity. The general Pix2Pix [14] framework takes generator with U-Net [28] architecture and performs down-sampling convolutions as many as possible in order to make the connection for remote pixels. For discriminator, it utilizes the PatchGAN architecture that maps input image to a Markov random field. Thus, the system has a big perception field and strengthens high-frequency correctness for generated samples. However, executing too many down-sampling convolutions causes too much loss of information so it leads to low fidelity of generated samples. To model enough perception field and remain much information, we combine a U-Net backbone with 4 ResBlocks [7] at bottleneck as our generator. For discriminator, we utilize a simplified PatchGAN architecture to enhance the perceptual quality. Moreover, the RAW to RGB mapping task is highly challenging due to lack of context information and data. To address these problems, we add a second decoder predicting saliency maps. It serves as an implicit data augmentation method which provides the system with low-level information. Overall, the proposed system aims at producing perceptually high-quality images based on saliency map-aided data augmentation.

3. Methodology

The architecture of the proposed system is shown in Figure 2. It contains an U-Net [28] structured generator and a Patch-based discriminator [14]. The generator produces RGB images and saliency maps given RAW file input. There are two decoders of the generator, while the shortcut connection is between encoder and RGB image decoder branch. The saliency map decoder is attached to each layer of RGB image decoder without sharing weights. The discriminator receives generated images or ground truth and maps them to a feature embedding. The details of the proposed architecture are presented in the following subsections.

3.1. Network architecture

There are two main techniques used in generator that improve the quality and accelerate convergence: ResNet [7] and U-Net [28]. The ResBlocks have been shown priority in many recent generative models [19, 37]. Instead of stacking few layers directly for underlying mapping, the

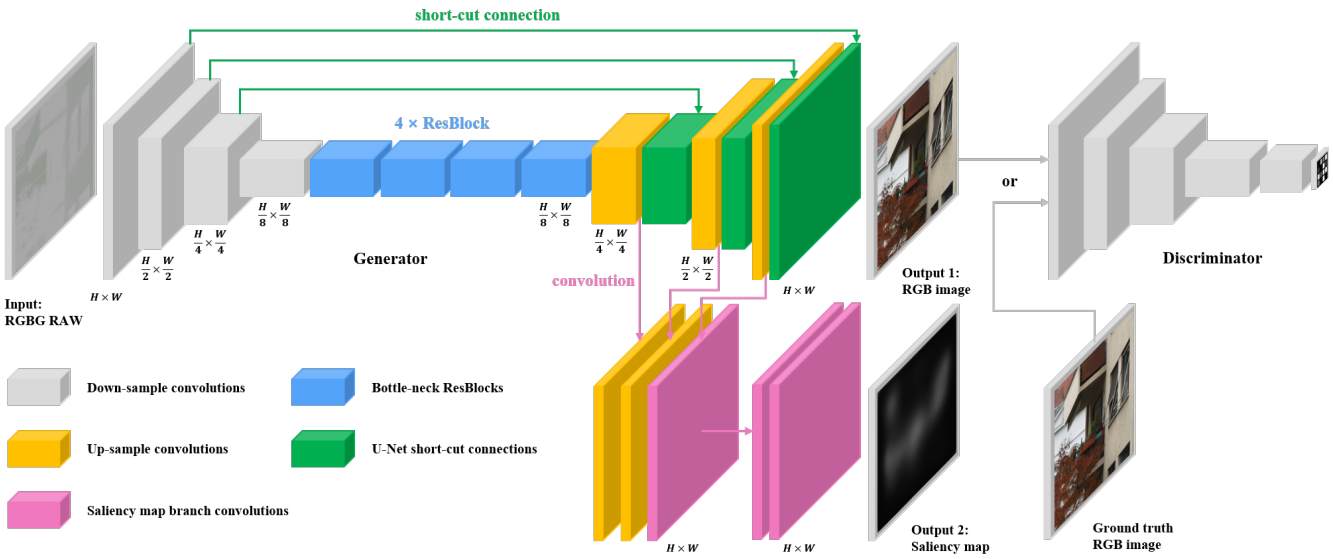

Figure 2. Architecture of the proposed RAW to RGB Generative Adversarial Network. The generator applies U-Net structure and receives a RAW file as input, finally generates a visible image with corresponding saliency map. The discriminator judges its input RGB image whether real or fake.

residual connection explicitly adds the input to the output. Normally, ResBlocks are embedded into the bottleneck part, which effectively extract the high-level semantics for constructing visible RGB images. Moreover, four ResBlocks obviously enhance the perceptual quality due to the enlarged perception field. The other design U-Net is to directly replicate each encoder block to decoder part with same shape, which effectively reduces gradient vanishing. The short-connection is only implemented in RGB image decoder. By combining the low-level details and decoder feature maps, it is easier to remain the edges for the output. To improve the perceptual quality, we further leverage a saliency map decoder for saliency detection. According to the visual saliency mechanism, we perform an element-wise product of two outputs to obtain the attention region and design an objective function based on it. For discriminator, we choose the PatchGAN architecture [14]. It maps input to an one-channel feature map by a series of convolutional layers. Compared with regular GAN that maps input to a scalar, it signifies which patch of input is real and fake. Furthermore, the system optimizes sensitive patches by tracing back the receptive field. However, these patches are overlapped to further enhance perceptual details and connections. To stabilize the training process, all the layers are Spectral Normalized [23] and LSGAN [21] critic is utilized.

3.2. Objectives

Since random initialization of GAN often leads to model collapse [33], it is significant to balance the training of both generator and discriminator. To facilitate and stabilize its convergence, the training process is divided into two stages. At first stage, we only train the generator with L1 loss in order to achieve high pixel accuracy. That means we perform a PSNR-oriented optimization for the system, which is also vital for stabilizing GAN training at second phase because the generator already produces relatively good results. The RGB image construction L1 loss and attention region L1 loss are defined as:

$$L_{RGB} = \mathbb{E}[||G_1(x) - y_1||_1], \tag{1}$$

$$L_A = \mathbb{E}[||G_1(x) \odot G_2(x) - y_1 \odot y_2||_1], \tag{2}$$

where $G_1(x)$, $G_2(x)$ are the output RGB image and saliency map, respectively. While y_1, y_2 are their corresponding ground truth. The operator $\odot$ means Hadamard product.

At second stage, we optimize the whole system by alternately training the generator and discriminator. To discriminate between the real images from the generated images, we use LSGAN critic. The loss function of both generator and discriminator are defined as:

$$L_G = \frac{1}{2}\mathbb{E}[(D(G_1(x)) - 1)^2], \tag{3}$$

$$L_D = \frac{1}{2}\mathbb{E}[(D(y_1) - 1)^2] + \frac{1}{2}\mathbb{E}[(D(G_1(x)))^2], \tag{4}$$

where D is the discriminator. Jointly minimizing equation 3 and 4 yields minimizing the Pearson χ^2 divergence between generated samples and ground truth.

Although L1 loss facilitates constructing images with less distortion, it leads to blurry details. To further enhance visual quality, perceptual loss [15] has been demonstrated to strengthen the high-frequency part and promote sharp edges [33]. Instead of directly measuring the pixel-level L1 loss, perceptual loss computes the distance between two images in high-level feature space. To match more semantic information, we utilize a deep layer of VGG-16 [30] network pre-trained on ImageNet. The feature map activated by ReLU [24] is very sparse, therefore we use output of convolutional layer to represent semantics. It is defined as:

$$L_p = \mathbb{E}[||\phi_l(G_1(x)) - \phi_l(y_1)||_1], \tag{5}$$

where $\phi_l(*)$ is the features of the $l-th$ layer of the pre-trained CNN. In our experiment, we use the $conv_{4\_3}$ layer of VGG-16 network pre-trained on ImageNet dataset.

The total loss function for generator at second stage is:

$$Loss = L_{RGB} + \lambda_A L_A + \lambda_G L_G + \lambda_p L_p \tag{6}$$

where the trade-off parameters λ_A, λ_G, and λ_p are empirically set to 0.5, 0.01, and 1, respectively.

3.3. Spectral normalization

As GAN architecture becomes more complicated, the initial loss definition [5] often leads to unstable training. Recently, extensive researches have leveraged many loss designs such as f-GAN [25], LSGAN [21], WGAN [1] for stabilizing learning process. However, the GAN framework is still sensitive to perturbation of input, therefore effective regularization methods are significant for generalizing inference of GAN. To restrict solution space, WGAN-GP [6] executes a well-designed weight decay technique when optimizing discriminator. Basically, it is relatively difficult to control regularization factor while gradient penalty may result in the trained model losing information. The other regularization method is adding noise to input and performing adversarial training [3]. Hence, the trained model is robust to perturbation of data. However, the generation quality is also affected. Instead of directly restricting data or gradient, Spectral Normalization [23] only adjusts the maximum eigenvalue for weights. It remains the original scale information of weight metrics thus lets GAN system meet 1-Lipschitz condition. Moreover, it does not cost numerous additional computational resources. By performing spectral normalization to each convolutional layer of discriminator, our system converges faster and more stable.

3.4. Two time-scale update rule (TTUR)

Based on derivation of original GAN [5], discriminator needs to be well trained to decrease Jensen-Shannon-divergence (JSD). Normally, It needs more training iterations for discriminator than generator. However, the convergence of GAN training is still hard to estimate. Heusel et al. [8] proposed a two time-scale update rule (TTUR) that executes individual learning rate for generator and discriminator. TTUR enhances the general performance of GAN and effectively prevent the mode collapse. At second stage, TTUR with Adam optimizer [16] have been applied to proposed system, which efficiently converges to a stationary local Nash equilibrium.

3.5. Implementation details

We use the training set of Zurich RAW2RGB (ZRR) dataset [11] to train the whole system. It contains 89000 paired images with a large diversity of contents, such as sky, buildings, cars, and streets. For each RGB image, we utilize pre-trained SalGAN to generate corresponding saliency map. It serves as an implicit data augmentation, which makes whole system more general and robust. All the training images are pre-processed to 224×224. The input RAW files and output RGB images are normalized to [-1, 1]. The output saliency map is rescaled to [0, 1], which conveniently represents attention region.

As aforementioned, the training process is divided into two stages. First, we train a PSNR-oriented generator only with L1 loss and attention loss for 10 epochs. The learning rate is fixed to 2×10^{-4}. At second stage, we train the generator and discriminator collaboratively for 30 epochs. While the initial learning rate for generator and discriminator are 1×10^{-4} and 4×10^{-4}, respectively. For optimization, we use Adam optimizer [16] with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and batch size equals to 4. The learning rates are halved every 10 epochs. The parameters of network are initialized using zero mean Gaussian distribution with standard deviation of 0.02. We perform reflection padding in the system to avoid border artifacts. LeakyReLU [20] activation function and Instance Normalization [31] is attached to each convolutional layer for both generator and discriminator except the output layers. When performing back propagation, the RGB image decoder branch is affected by all the objectives. We implement our system with PyTorch framework and train it on two NVIDIA GeForce GTX 1080 Ti GPUs. It takes approximately 4 days to complete the whole 40 epochs.

4. Experiment

We evaluated both pixel fidelity and perceptual realism of proposed GAN system for RAW to RGB mapping on ZRR validation set (2139 pairs). Quantitative and qualitative results for generated RGB images and full resolution photographs are presented in next section. We also reported our results of AIM 2019 RAW to RGB mapping challenge on ZRR testing set (2462 pairs). Both validation and testing images are cropped to 224×224 for evaluation. Finally, we discussed the failure cases and future work of proposed system.

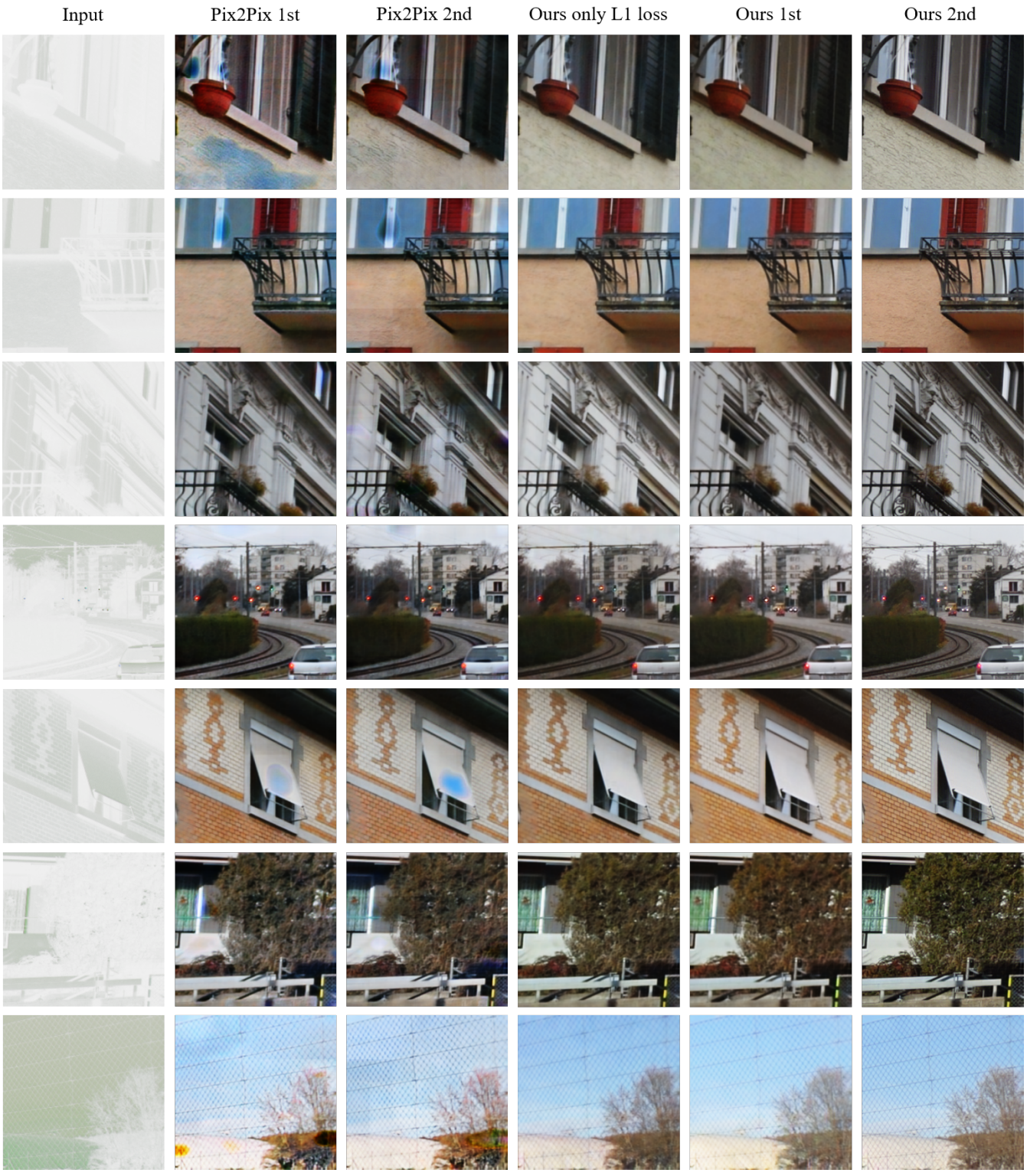

Figure 3. Comparison of generated results of different models. The first column shows the input RAW files. The second to last columns represent the generated RGB photos by Pix2Pix (first stage), Pix2Pix (second stage), proposed architecture (only trained with L1 loss for 40 epochs), proposed approach (first stage), proposed approach (second stage), respectively.

4.1. Evaluation metrics

In order to effectively assess the generation quality, the two metrics Peak Signal-to-Noise Ratio (PSNR) and Mean Opinion Score (MOS) are adopted. PSNR is widely used to represent the ratio between maximum power of signal and power of noise that affects the fidelity quality. In order to adapt to different dynamic range, PSNR is expressed in terms of the logarithmic decibel scale. A generated image with high PSNR means there is lower probability to be noisy and distorted. It is defined as:

$$PSNR = 10 \cdot log_{10}(\frac{255^2 \cdot W \cdot H}{\sum_{w=1}^{W} \sum_{h=1}^{H} [I(w,h) - K(w,h)]^2}), \tag{7}$$

where I and K represent generated image and ground truth, respectively. Their width and height are given by W and H.

However, there is no strong relationship between PSNR and visual quality from human opinion. To measure the perceptual experience, MOS is commonly utilized in image quality assessment. As a subjective metric, MOS is a single rational number, which is given by human assessors. Compared with PSNR, MOS is more reasonable to represent overall perceptual quality of the system.

4.2. Quantitative analysis

To demonstrate that proposed method produces perceptually plausible RGB photos, we trained a Pix2Pix framework [14] with same settings (two training stages, learning rate, batch size, and epochs). We compared the average PSNR using generated 2139 samples by Pix2Pix framework and proposed approach on ZRR validation set. As shown in Table 1, the proposed approach improves general Pix2Pix in about 20%. The saliency map prediction branch of proposed approach enhances the system to learn visual saliency information although training set is limited. Moreover, saliency map provides more low-level details for system to construct RGB images. On the other hand, we added perceptual loss and adversarial loss at second stage. The system gets an improvement of about 0.5 with respect to first stage. We also validated the proposed system trained only with L1 loss for 40 epochs. The PSNR equals to 22.244904, which means the system trained with full objectives outperforms trained only with L1 loss.

For the testing phase, the PSNR is 21.91 on ZRR testing set (2462 pairs) [11]. We also evaluated the average time for our model to process the testing images of 224×224 resolution. We ran the model on test machine with Intel Core i9-9900K CPU, @ 3.60 GHz, 8 cores and single NVIDIA GeForce GTX 1080 Ti GPU. The mean value of overall 20 computations is adopted. For each input image, it takes 0.03571s (27.98 images/second). The proposed system is suitable for real-time usage.

Method	Pix2Pix 2nd	Ours 1st	Ours 2nd
PSNR	19.491088	22.067571	22.455825
SSIM	0.727142	0.798382	0.798674

Table 1. Comparison of proposed method with Pix2Pix. The '1st' and '2nd' represents the two training stages.

Figure 4. Generated full resolution images.

Figure 5. Failure examples. The proposed method is sensitive to small details in first row. And second row shows unreasonable blurry results.

4.3. Qualitative analysis

We randomly select some generated samples from Pix2Pix and proposed system, as shown in Figure 3. To compare the perceptual effect, we illustrate some patches for two training stages from both Pix2Pix [14] and proposed approach. Because both systems are trained only with L1 loss at first stage, the generated samples are obviously more blurry than second stage. There is artifact in the images produced by Pix2Pix due to Batch Normalization [13]. Moreover, we show the results produced by proposed architecture trained only with L1 loss for 40 epochs. Note that, our proposed system is optimized by whole objectives for last 30 epochs. It demonstrates that adversarial training and perceptual loss indeed enhance perceptual quality.

4.4. Full resolution results

As proposed system is a fully convolutional architecture, it is possible to operate input images with any resolution although it is trained by small patches (224×224 resolution). However, restricted by memory, it is difficult to generate a full resolution image (around 2000×1500 resolution) in one forward process on a single 1080 Ti GPU. Therefore, we perform a sliding window method to produce full resolution RGB photos patch-by-patch (maximum 768×768 resolution for small patch). It takes approximately 0.5 seconds to render a full resolution image on same test machine. The result RGB images are shown in Figure 4.

4.5. Failure cases

Figure 5 shows some examples where proposed method fails to generate high-quality results. For the CNN based models, reconstructing patches with too many details is troublesome. For example, there is a little color bleeding effects in coincident parts of branches and sky. We also illustrate some examples that are unreasonably blurry. In the future, multiple training datasets and more effective objective functions can contribute to better results.

5. Conclusion

In this paper, we address the problem of RAW to RGB mapping using saliency map-aided Generative Adversarial Network. The proposed architecture is trained in two stages for fidelity and perceptual quality. In addition, we propose an implicit saliency map data augmentation method technique to enhance the joint RGB image and saliency map prediction. Experimental results show that our approach has strong ability to construct perceptually plausible photographs even training data is limited.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[2] Bryce E Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. 2017.

[4] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[9] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3277–3285, 2017.

[10] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 691–700, 2018.

[11] Andrey Ignatov, Radu Timofte, et al. Aim 2019 challenge on raw to rgb mapping: Methods and results. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[12] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG)*, 35(4):110, 2016.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2018.

[17] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016.

[18] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 689–694. ACM, 2004.

[19] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. 2019.

[20] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models.

In *International Conference on Machine Learning*, page 3, 2013.

[21] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[22] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[23] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[24] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, pages 807–814, 2010.

[25] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016.

[26] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Cristian Canton Ferrer, Jordi Torres, Kevin McGuinness, and Noel E O'Connor. Salgan: Visual saliency prediction with adversarial networks. In *CVPR Scene Understanding Workshop (SUNw)*, 2017.

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[31] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[32] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[33] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *European Conference on Computer Vision*, pages 0–0, 2018.

[34] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 277–280. ACM, 2002.

[35] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Texturegan: Controlling deep image synthesis with texture patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2018.

[36] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[37] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.