Self-Supervised Colorization towards Monochrome-Color Camera Systems Using Cycle CNN

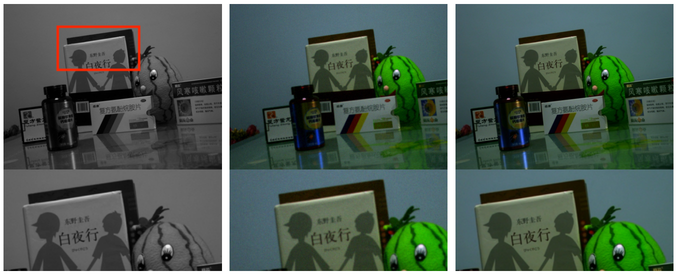
Xuan Dong, Chang Liu, Weixin Li*, Xiaoyan Hu, Xiaojie Wang, Yunhong Wang, *Fellow, IEEE*

***Abstract*—Colorization in monochrome-color camera systems aims to colorize the gray image $\mathbf{I_G}$ from the monochrome camera using the color image $\mathbf{R_C}$ from the color camera as reference. Since monochrome cameras have better imaging quality than color cameras, the colorization can help obtain higher quality color images. Related learning based methods usually simulate the monochrome-color camera systems to generate the synthesized data for training, due to the lack of ground-truth color information of the gray image in the real data. However, the methods that are trained relying on the synthesized data may get poor results when colorizing real data, because the synthesized data may deviate from the real data. We present a self-supervised CNN model, named Cycle CNN, which can directly use the real data from monochrome-color camera systems for training. In detail, we use the Weighted Average Colorization (WAC) network to do the colorization twice. First, we colorize $\mathbf{I_G}$ using $\mathbf{R_C}$ as reference to obtain the first-time colorization result $\mathbf{I_C}$. Second, we colorize the de-colored map of $\mathbf{R_C}$, i.e. $\mathbf{R_G}$, using the concatenated image of $\mathbf{I_G}$ and Cb/Cr channels of the first-time colorization result $\mathbf{I_C}$, i.e. $\mathbf{I}_{\mathbf{C}}^{Cb}$ and $\mathbf{I}_{\mathbf{C}}^{Cr}$, as reference to obtain the second-time colorization result $\mathbf{R}_{\mathbf{C}}^{'}$. In this way, for the second-time colorization result $\mathbf{R}_{\mathbf{C}}^{'}$, we use the Cb and Cr channels of the original color map $\mathbf{R_C}$ as ground-truth and introduce the cycle consistency loss to push $\mathbf{R}_{\mathbf{C}}^{'Cb/Cr} \approx \mathbf{R}_{\mathbf{C}}^{Cb/Cr}$. Also, for the Y channel of the first-time colorization result $\mathbf{I}_{\mathbf{C}}^{Y}$, we propose the Global Curve Adjustment (GCA) network and the structure similarity loss to encourage the structure similarity between $\mathbf{I}_{\mathbf{C}}^{Y}$ and $\mathbf{I_G}$. In addition, we introduce a spatial smoothness loss within the WAC network to encourage spatial smoothness of the colorization result. Combining all these losses, we could train the Cycle CNN using the real data in the absence of the ground-truth color information of $\mathbf{I_G}$. Experimental results show that we can outperform related methods largely for colorizing real data.**

***Index Terms*—Weighted Average Colorization, Global Curve Adjustment, Cycle Consistency, Structure Similarity, Spatial Smoothness.**

I. Introduction

With the increasing use of monochrome-color multi-lens camera systems in high-end smart phones, e.g. Huawei P30, Mate30, etc., the colorization problem within these systems is attracting more and more attentions from the academic and industrial communities.



(a) Input pair of gray image $\mathbf{I_G}$ and color image $\mathbf{R_C}$. (b) Our colorization result $\mathbf{I}_{\mathbf{C}}^{*}$.

Fig. 1. The gray image $\mathbf{I_G}$ and color image $\mathbf{R_C}$ in the input pair are shot by the monochrome and color cameras, respectively. By directly using these real data for training, our algorithm learns to colorize $\mathbf{I_G}$ using $\mathbf{R_C}$ as reference.

As shown in Fig. 1, colorization in monochrome-color camera systems aims to colorize the gray image $\mathbf{I_G}$ from the monochrome camera using the color image $\mathbf{R_C}$ from the color camera as reference. Between the monochrome and color cameras, there exist different hardwares, e.g. the color filter array, and different software modules, e.g. white balance, demosaic, etc. As a result, on the one hand, the monochrome camera has better light efficiency [1], [2] than the color camera, and thus the gray image $\mathbf{I_G}$ has higher quality, i.e. signal-noise ratio, than the color image $\mathbf{R_C}$. This motivates researchers to do the colorization so as to get higher quality color images using the monochrome-color camera systems. On the other hand, the pair of gray and color images have different luminance, blur, noises, etc., which bring difficulties for the colorization.

Among existing methods for colorization within the monochrome-color camera system, some are traditional hand-crafted methods, e.g. [1]. With the successful use of deep learning in various computer vision problems, some deep learning based methods, e.g. [2] are proposed recently, which have shown to be able to obtain higher accuracy than the traditional ones. However, in the deep learning methods, e.g. [2], the models usually need ground-truth color information of the input gray images for training. Due to the lack of ground-truth color information in the real data, as shown in Fig. 2, current methods, e.g. [1], [2], synthesize data to simulate the real data from the monochrome-color camera system. However, the degradation models for synthesizing the data may deviate from the ones in real imaging systems within the monochrome and color cameras. Thus, the synthesized data could hardly simulate the real data perfectly. As a result, the deep learning methods, which are trained relying on the synthesized data, may have very poor results when colorizing

*Corresponding author. Email: weixinli@buaa.edu.cn.

Xuan Dong, Chang Liu, Xiaoyan Hu and Xiaojie Wang are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China.

Weixin Li and Yunhong Wang are with the School of Computer Science and Engineering, Beihang University, Beijing, China.

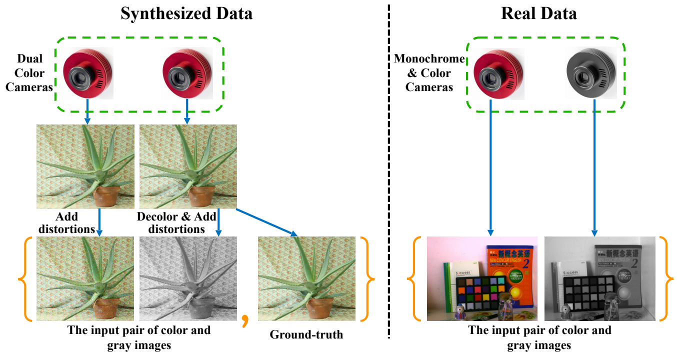

Synthesized Data
Real Data
Dual Color Cameras
Monochrome & Color Cameras
Add distortions
Decolor & Add distortions
The input pair of color and gray images
Ground-truth
The input pair of color and gray images

Fig. 2. How the synthesized data and the real data are obtained. The real data are the pair of gray and color images shot from the monochrome-color camera system. The synthesized data are the pair of gray and color images that are synthesized using a pair of color images from the dual-color camera system.

the real data.

To overcome this limitation, in this paper, we propose a self-supervised colorization model and aim to directly use the real data from the monochrome-color camera system for training.

Our insight is based on the property of cycle colorization consistency. As shown in Fig. 3, when we do the colorization twice, i.e. firstly colorizing $\mathbf{I}_{\mathbf{G}}$ using $\mathbf{R}_{\mathbf{C}}$ as reference and secondly colorizing the gray map of $\mathbf{R}_{\mathbf{C}}$, i.e. $\mathbf{R}_{\mathbf{G}}$, using the obtained first-time colorization result $\mathbf{I}_{\mathbf{C}}$ as reference, the second-time colorization result $\mathbf{R}_{\mathbf{C}}^{'}$ should arrive back at $\mathbf{R}_{\mathbf{C}}$. Thus, to train the colorization model, we can use $\mathbf{R}_{\mathbf{C}}$ as the ground-truth for $\mathbf{R}_{\mathbf{C}}^{'}$, and $\mathbf{I}_{\mathbf{G}}$ as the ground-truth for the Y channel of $\mathbf{I}_{\mathbf{C}}$, i.e. $\mathbf{I}_{\mathbf{C}}^{Y}$. In this way, the training data are from the real data shot by the monochrome-color camera system and we do not need any synthesized data at all.

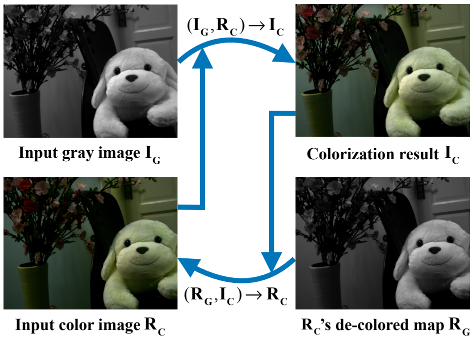

$(I_G, R_C) \rightarrow I_C$
Input gray image I_G
Colorization result I_C
$(R_G, I_C) \rightarrow R_C$
Input color image R_C
R_C's de-colored map R_G

Fig. 3. The insight of cycle colorization consistency. When doing the colorization twice, i.e. firstly $(\mathbf{I}_{\mathbf{G}}, \mathbf{R}_{\mathbf{C}}) \rightarrow \mathbf{I}_{\mathbf{C}}$ and secondly $(\mathbf{R}_{\mathbf{G}}, \mathbf{I}_{\mathbf{C}}) \rightarrow \mathbf{R}_{\mathbf{C}}^{'}$, the second-time colorization result $\mathbf{R}_{\mathbf{C}}^{'}$ should arrive back at $\mathbf{R}_{\mathbf{C}}$.

Based on this insight, we propose a self-supervised CNN model, named Cycle CNN. As shown in Fig. 4, we propose a Weighted Average Colorization (WAC) network to do the colorization twice, i.e. firstly colorizing $\mathbf{I}_{\mathbf{G}}$ using $\mathbf{R}_{\mathbf{C}}$ as reference and secondly colorizing $\mathbf{R}_{\mathbf{G}}$ using the concatenated image of $\mathbf{I}_{\mathbf{G}}$ and Cb/Cr channels of the first-time colorization result $\mathbf{I}_{\mathbf{C}}$, i.e. $\mathbf{I}_{\mathbf{C}}^{Cb}$ and $\mathbf{I}_{\mathbf{C}}^{Cr}$, as reference. We do horizontal flips for the inputs and outputs of the second-time colorization so as to enable the WAC network to perform the second-time colorization without changing any model structure.

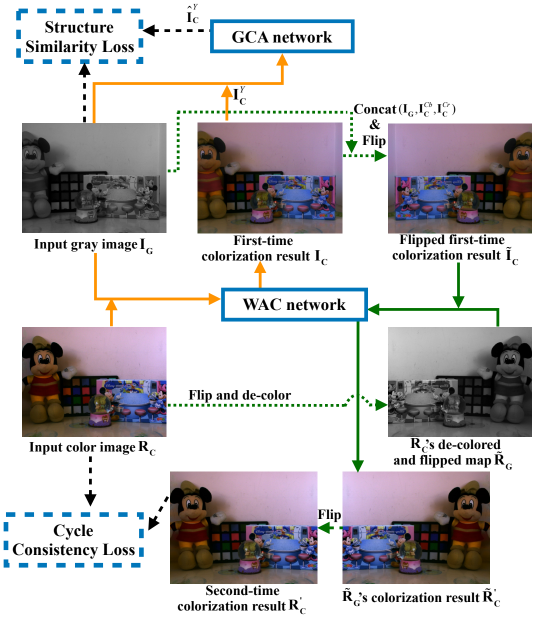

Structure Similarity Loss
$\hat{I}_C^Y$
GCA network
I_C^Y
Concat (I_G, I_C^{Cb}, I_C^{Cr}) & Flip
Input gray image I_G
First-time colorization result I_C
Flipped first-time colorization result $\tilde{I}_C$
WAC network
Flip and de-color
Input color image R_C
R_C's de-colored and flipped map $\tilde{R}_G$
Cycle Consistency Loss
Flip
Second-time colorization result $R_C^{'}$
$\tilde{R}_G$'s colorization result $\tilde{R}_C^{'}$

Fig. 4. The overall structure of our Cycle CNN model.

We design three losses, namely the structure similarity loss, the cycle consistency loss and the spatial smoothness loss, for training the proposed Cycle CNN model, as shown in Fig. 4. First, the structure similarity loss aims to measure the structure similarity between $\mathbf{I}_{\mathbf{C}}^{Y}$ and $\mathbf{I}_{\mathbf{G}}$. Due to different exposure settings, e.g. shutter speed, ISO, etc., and camera response functions (CRFs) [3] between the monochrome camera and the color camera, the corresponding pixels between $\mathbf{I}_{\mathbf{C}}^{Y}$ (which is from $\mathbf{R}_{\mathbf{C}}$ that is shot by the color camera) and $\mathbf{I}_{\mathbf{G}}$ (which is shot by the monochrome camera) usually have different luminance. And the structure similarity loss should avoid the affects of the luminance differences. The luminance differences caused by different exposure and CRFs can be adjusted by a global curve, which is widely used in High Dynamic Range imaging [3]. So, to overcome this challenge, we propose the Global Curve Adjustment (GCA) network which estimates a global adjustment curve T and uses it to perform global curve adjustment for $\mathbf{I}_{\mathbf{C}}^{Y}$ to get $\widehat{\mathbf{I}}_{\mathbf{C}}^{Y}$, so that the luminance differences between $\widehat{\mathbf{I}}_{\mathbf{C}}^{Y}$ and $\mathbf{I}_{\mathbf{G}}$ caused by different exposure and CRFs are minimized. Then, we use the difference between $\widehat{\mathbf{I}}_{\mathbf{C}}^{Y}$ and $\mathbf{I}_{\mathbf{G}}$ as the structure similarity loss between $\mathbf{I}_{\mathbf{C}}^{Y}$ and $\mathbf{I}_{\mathbf{G}}$. Second, the cycle consistency loss is designed to encourage the similarity of the Cb and Cr channel maps between $\mathbf{R}_{\mathbf{C}}^{'}$ and $\mathbf{R}_{\mathbf{C}}$. Last, the spatial smoothness loss aims to encourage the spatial smoothness of the colorization result. We also use a refinement CNN to refine the Cb and Cr channels of $\mathbf{I}_{\mathbf{C}}$, i.e. $\mathbf{I}_{\mathbf{C}}^{Cb}$ and $\mathbf{I}_{\mathbf{C}}^{Cr}$, with $\mathbf{I}_{\mathbf{G}}$ as guidance to get the refined Cb and Cr channels $\mathbf{I}_{\mathbf{C}}^{*Cb}$ and $\mathbf{I}_{\mathbf{C}}^{*Cr}$ and concatenate $\mathbf{I}_{\mathbf{G}}$, $\mathbf{I}_{\mathbf{C}}^{*Cb}$ and $\mathbf{I}_{\mathbf{C}}^{*Cr}$ to get the final result $\mathbf{I}_{\mathbf{C}}^{*}$.

Experimental results show that we can outperform related methods largely for the real data from the monochrome-color camera system.

Our main contributions include 1) the self-supervised Cycle CNN which can be trained using real data from the monochrome-color camera systems without any human annotation, 2) the GCA network and the new structure similarity loss for measuring the quality of the first-time colorization result, and 3) the combination of the newly proposed modules with the modules from our previous work in [4], including the WAC network, the cycle consistency loss for measuring the second-time colorization result, and the spatial smoothness loss for spatial smoothness of the colorization result.

Our framework is an extension of our previous work in [4]. In comparison with [4], the differences and improvements are as follows: 1) To measure the quality of the first-time colorization result, the previous work in [4] builds a dataset by image registration, manually selecting and cropping well-registered regions and random warping, and pre-trains the structure similarity loss based on the dataset. It has two limitations. First, when using a new monochrome-color camera system, the dataset has to be re-built again and the pre-trained loss has to be re-trained as well. This increases complexity and human labor a lot. Second, the pre-trained loss may not be accurate for practical data because the training samples in the dataset may deviate from the practical samples. To overcome these limitations, in this paper, we propose the GCA network and the new structure similarity loss to make the whole Cycle-CNN an end-to-end and self-supervised framework. 2) We provide more quantitative and qualitative experimental results in comparison with the state-of-the-art algorithms and we also provide more ablation study results in this paper.

II. Related Works

The existing colorization tasks can be divided into five kinds, i.e. automatic colorization, text-based colorization, scribble-based colorization, reference-based colorization, and monochrome-color dual-lens colorization.

In automatic colorization, the input is only a single gray image and the algorithms need to automatically colorize it without any reference. Recent deep learning based methods, e.g. [5], [6], [7], [8], [9], and [10], make great progress to solve this problem. However, these methods are not proper for our problem because they fail to make use of the color image from the color camera, which contains much useful color information for colorizing the gray image from the monochrome camera.

In text-based colorization, Bahng et al. [11] introduce a manually curated dataset, called Palette-and-Text (PAT), and propose the Text2Colors model which consists of two conditional generative adversarial networks (GANs), i.e. the text-to-palette generation network to capture the semantics of the text input and produce relevant color palettes and the palette-based colorization network to colorize a grayscale image using the generated color palette. Manjunatha et al. [12] propose a language-conditioned colorization method which allows end users to manipulate the process of image colorization by feeding in different captions. Kim et al. [13] propose a GAN based line art colorization method which takes as input a grayscale line art and color tag information and produces the colorization result. These methods are not suitable for our problem because the text information is not available in the monochrome-color camera system.

In the scribble-based colorization task, the input includes a single gray image and several color scribbles which are drawn by humans. And the methods, e.g. [14] and [15], use the color scribbles as guidance to propagate the colors to the whole image. These methods are not suitable for our problem because there exist no scribbles in the monochrome-color camera system.

In the reference-based colorization task, the input includes an input gray image and a reference color image. Different from our problem, the reference image is shot in different locations and/or at different time and the contents within the pair of images just share similar semantics. Because the inputs are different from ours, the methods, e.g. [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], usually firstly search semantically similar pixels between the images and then propagate the colors of the matching pixels to the whole image. Welsh et al. [16] assume that pixels with the same grayscale intensity will have the same color, and use the luminance value as the feature to search for matching pixels. Ironi et al. [17] use discrete cosine transform coefficients as the feature to search sparse matching pixels, copy the color of matching pixels for pixels in high confidence regions and then colorize pixels in low confidence regions by color propagation [15]. Gupta et al. [18] extract features of superpixels by averaging feature values of all pixels among each superpixel, search for matching pixels by feature matching and use space voting for spatial consistency. Furusawa et al. [19] propose a reference-based colorization algorithm for colorizing manga images. The assumption for manga images are not always correct for general images. Thus, their results are not always good enough for solving our problem. He et al. [21], [22], Zhang et al. [26], Xu et al. [23] and Lu et al. [24] propose deep learning based algorithms for image and video colorization. But, they usually assume the pair of images are visually very different but semantically similar. Due to different assumptions from our problem, they do not consider locality and spatial smoothness and the proposed methods usually minimize the semantic differences. Due to different assumptions and goals, their results are not always faithful to the correct colors. Lee et al. [25] make use of internal attention mechanism and dense semantic correspondence to propose a reference-based colorization algorithm for sketch images. But the results are poor for our problem because the sketch images are much different from monochrome images and the combination of the similarity loss, perceptual loss, style loss, and adversarial loss is not proper for the monochrome-color dual-lens colorization task. In addition, the existing learning-based methods for reference-based colorization are supervised methods, while in the colorization problem for real monochrome-color camera systems, there are no annotation data for the supervised methods to train or fine-tune.

The monochrome-color dual-lens colorization task can be seen as a special case of reference-based colorization. Jeon et al. [1] propose a stereo matching method to search for best-matching pixels, and correct colors in occlusion regions by applying spatial consistency of neighboring pixels over the whole image. But the accuracy for the stereo matching is not high all the time, especially in occlusion regions, and the wrongly estimated correspondences will lead to wrong colorization results. Dong et al. [2], [27] proposed a deep CNN for solving this problem and make use of the cycle consistency property to propose the colorization quality assessment module for the colorization results. However, they are still traditional supervised methods and rely on synthesized data to train the model. As discussed in the Introduction Section, the real data are quite different from the synthesized data, and their performance on the real data decreases largely.

Like sparseness, smoothness, etc., cycle consistency is also a marvelous and general property and can be utilized for solving different vision problems, e.g. image translation [28], visual tracking [29], super resolution [30], image quality assessment [27], etc. In this paper, we make use of it for solving the self-supervised colorization problem towards real monochrome-color camera systems.

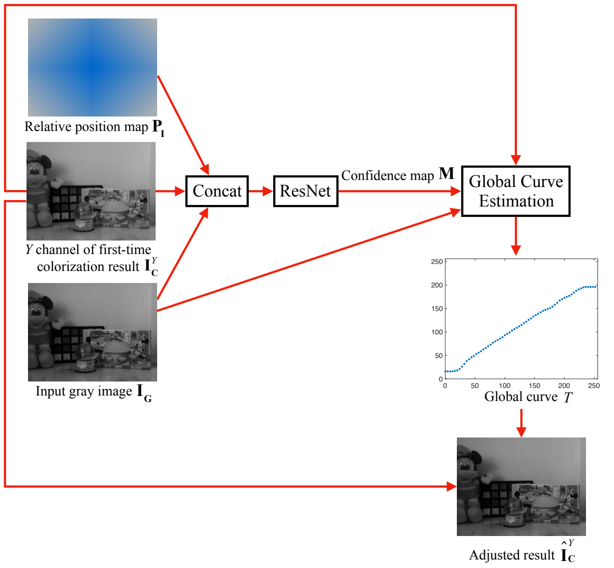

Relative position map $\mathbf{P_I}$
Y channel of first-time colorization result $\mathbf{I}_\mathbf{C}^Y$
Input gray image $\mathbf{I_G}$
Concat
ResNet
Confidence map $\mathbf{M}$
Global Curve Estimation
250
200
150
100
50
0
0
50
100
150
200
250
Global curve T
Adjusted result $\hat{\mathbf{I}}_\mathbf{C}^Y$

Fig. 5. The structure of our GCA network.

III. Method

A. Overview

As shown in Fig. 4, our Cycle-CNN framework does the colorization using the Weighted Average Colorization (WAC) network twice. In the first-time colorization, we colorize the input gray image $\mathbf{I_G} \in \mathbb{R}^{h\times w}$ using the color image $\mathbf{R_C} \in \mathbb{R}^{h\times w\times 3}$ as reference, where h and w denote the height and width of the image, respectively. After getting the first-time colorization result $\mathbf{I_C} \in \mathbb{R}^{h\times w\times 3}$, in the second-time colorization, we do horizontal flipping for the de-colored image of $\mathbf{R_C}$, named $\mathbf{R_G} \in \mathbb{R}^{h\times w}$, and the concatenated image of $\mathbf{I_G}$, $\mathbf{I}_\mathbf{C}^{Cb}$ and $\mathbf{I}_\mathbf{C}^{Cr}$ and then feed the flipped images into the WAC network. The result is then flipped again to get the second-time colorization result $\mathbf{R}_\mathbf{C}^{'} \in \mathbb{R}^{h\times w\times 3}$. We do the three flip operations because the WAC network always searches colors of pixels in the range of (j, i) to $(j, i+d-1)$ in the reference image for each pixel (j, i) in the input gray image, and, in the second-time colorization, the corresponding pixels in the concatenated image of $(\mathbf{I_G}, \mathbf{I}_\mathbf{C}^{Cb}, \mathbf{I}_\mathbf{C}^{Cr})$ locate in the opposite search range, i.e. from (j, i) to $(j, i-d+1)$. By doing the flip operations, we can enable the WAC network to perform the second-time colorization without changing any model structure.

For the first-time colorization result, we design the structure similarity loss to encourage the structure similarity between the Y channel of $\mathbf{I_C}$, i.e. $\mathbf{I}_\mathbf{C}^Y \in \mathbb{R}^{h\times w}$, and $\mathbf{I_G}$. Due to the luminance differences of the corresponding pixels between $\mathbf{I}_\mathbf{C}^Y$ and $\mathbf{I_G}$ which is caused by different exposure settings and camera response functions (CRFs) of the monochrome and color cameras, we propose the Global Curve Adjustment (GCA) network. As shown in Fig. 5 ,first, we use $\mathbf{I}_\mathbf{C}^Y$, $\mathbf{I_G}$ and the relative position map $\mathbf{P_I}$ to estimate the confidence map $\mathbf{M}$, i.e. how much each pixel contributes to the estimation of the global curve T. Second, we use $\mathbf{I}_\mathbf{C}^Y$, $\mathbf{I_G}$ and $\mathbf{M}$ to estimate the global curve T. Third, we use the estimated global curve T to adjust $\mathbf{I}_\mathbf{C}^Y$ to get $\widehat{\mathbf{I}}_\mathbf{C}^Y \in \mathbb{R}^{h\times w}$ so that the luminance differences between $\widehat{\mathbf{I}}_\mathbf{C}^Y$ and $\mathbf{I_G}$ are minimized. And the difference between $\widehat{\mathbf{I}}_\mathbf{C}^Y$ and $\mathbf{I_G}$ is then calculated and used as the structure similarity loss between $\mathbf{I}_\mathbf{C}^Y$ and $\mathbf{I_G}$.

For the second-time colorization result, we design the cycle consistency loss to encourage the similarity of the Cb/Cr maps between the second-time colorization result $\mathbf{R}_\mathbf{C}^{'}$ and the reference color image $\mathbf{R_C}$. The spatial smoothness loss is also designed to encourage the spatial smoothness of the colorization result.

We also use the refinement CNN to refine the Cb and Cr channels of the first-time colorization result $\mathbf{I_C}$, i.e. $\mathbf{I}_\mathbf{C}^{Cb}$ and $\mathbf{I}_\mathbf{C}^{Cr}$, with $\mathbf{I_G}$ as guidance to get the refined Cb and Cr channels $\mathbf{I}_\mathbf{C}^{*Cb}$ and $\mathbf{I}_\mathbf{C}^{*Cr}$ and concatenate $\mathbf{I_G}$, $\mathbf{I}_\mathbf{C}^{*Cb}$ and $\mathbf{I}_\mathbf{C}^{*Cr}$ to get the final result $\mathbf{I}_\mathbf{C}^{*}$.

The training data are the real data captured from monochrome-color camera systems, as shown in Fig. 2. And the whole self-supervised system does not need any extra synthesized data at all.

B. WAC Network

Within the WAC network, the input is the pair of gray and reference color images $\mathbf{A_G} \in \mathbb{R}^{h\times w}$ and $\mathbf{B_C} \in \mathbb{R}^{h\times w\times 3}$, and the output is the colorization result $\mathbf{A_C} \in \mathbb{R}^{h\times w\times 3}$, as shown in Fig. 6. First, we extract the deep features of the input images, i.e. $\mathbf{F_A}$ and $\mathbf{F_B}$. Next, $\mathbf{F_A}$ and $\mathbf{F_B}$ are used for building the 4-D feature volume $\mathbf{V^F}$ with the search range d. $\mathbf{V^F}$ is then fed into the 3-D U-Net to learn the 3-D weight volume $\mathbf{V^W} \in \mathbb{R}^{h\times w\times d}$. And, for each pixel (j, i), the colorization result $\mathbf{A_C}$ is obtained by the weighted average operation between $\mathbf{V^W}$ and $\mathbf{B_C}$, i.e.

$$\mathbf{A}_\mathbf{C}^c(j, i) = \sum_{k=0}^{d-1} \mathbf{V^W}(j, i, k)\mathbf{B}_\mathbf{C}^c(j, i+k), \qquad (1)$$

where $c \in \{Y, Cb, Cr\}$, and the search range d of candidate pixels for each pixel (j, i) is defined as the pixels with the same vertical position, i.e. j, and the horizontal positions range from i to $i+d-1$, where the hyper-parameter d controls the maximum disparity. The search range is within the same line because the dual-lens of phones are calibrated and the corresponding pixels should be in the same line but different columns due to disparity. Pixels in the defined range have high probability to provide correct colors. $\mathbf{V^W}(j,i,k)$ is the weight values between pixel (j,i) of the input gray image and pixel $(j,i+k)$ of the reference image, and the weight volume $\mathbf{V^W}$ contains the weight values of all pixels and their candidate pixels.

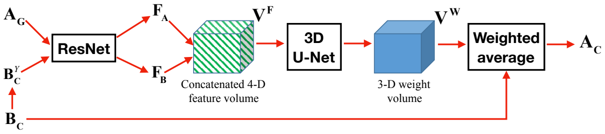


Fig. 6. The WAC network that colorize the given input gray image $\mathbf{A_G}$ using the input color image $\mathbf{B_C}$ as reference to obtain the colorization result $\mathbf{A_C}$.

C. GCA Network

As shown in Fig. 5, for the given pair of image $\mathbf{I_C^Y}$ and $\mathbf{I_G}$, our goal is to estimate the global curve T, and use it to perform global curve adjustment for $\mathbf{I_C^Y}$ to get the adjustment result $\widehat{\mathbf{I}}_\mathbf{C}^Y$ by

$$\widehat{\mathbf{I}}_\mathbf{C}^Y(j,i) = T(\mathbf{I_C^Y}(j,i)), \tag{2}$$

so that the luminance difference between $\widehat{\mathbf{I}}_\mathbf{C}^Y$ and $\mathbf{I_G}$ that are caused by different exposure and CRFs is minimized.

A global curve T ideally contains 256 nodes in the dynamic range of $[0,255]$. However, estimating all the 256 nodes will lead to heavy costs on the GPU memory. So we choose to estimate S sparse nodes with the stride of s. In this paper, we set $S = 64$ and thus $s = 4$. The input intensity of the sparse nodes is x_m, where x_1 to x_S are $[0,4,8,12,...,252]$, and the corresponding output intensity is y_m. An example of the estimated global curve T is shown in Fig. 5. Thus, for any given input intensity x, we estimate the corresponding output intensity $T(x)$ using the global curve T with sparse nodes by

$$T(x) = \frac{\sum\limits_{m=1}^{S} D_T(x,x_m)y_m}{\sum\limits_{m=1}^{S} D_T(x,x_m)}, \tag{3}$$

where $D_T(x,x_m)$ measures the Gaussian distance between any given input intensity x and the input intensity of each sparse node x_m. We define $D_T(x,x_m)$ as

$$D_T(x,x_m) = e^{-\frac{(x-x_m)^2}{2\sigma^2}}, \tag{4}$$

where σ is set to 5 in this paper.

For the given pair of image $\mathbf{I_C^Y}$ and $\mathbf{I_G}$, estimating the corresponding global curve T equals to estimating the output intensity y_m of x_m at each node. And we estimate y_m by the weighted average over all pixels of $\mathbf{I_G}$, i.e.

$$y_m = \frac{\sum\limits_{j,i} \mathbf{W_{GCA}}(j,i,m)\mathbf{I_G}(j,i)}{\sum\limits_{j,i} \mathbf{W_{GCA}}(j,i,m)}. \tag{5}$$

The GCA weight values of pixel (j,i) at node m, i.e. $\mathbf{W_{GCA}}(j,i,m)$, is defined as

$$\mathbf{W_{GCA}}(j,i,m) = D_T(\mathbf{I_C^Y}(j,i),x_m)\mathbf{M}(j,i), \tag{6}$$

where $D_T(\mathbf{I_C^Y}(j,i),x_m)$ measures the Gaussian distance between the intensity $\mathbf{I_C^Y}(j,i)$ and the input intensity at node m, i.e. x_m, and $\mathbf{M}(j,i)$ is the learned confidence value between $\mathbf{I_C^Y}$ and $\mathbf{I_G}$ at pixel (j,i), which shows how much pixel (j,i) contributes to the estimation of the global curve T. To learn the confidence map $\mathbf{M}$, we use $\mathbf{I_C^Y}$, $\mathbf{I_G}$ and the relative position map $\mathbf{P_I} \in \mathbb{R}^{h\times w\times 2}$ as the inputs. The relative position map $\mathbf{P_I}$ consists of the distance to the image boundary at the vertical direction and the horizontal direction of each pixel (j,i), i.e.

$$\mathbf{P_I}(j,i,1) = \min(j, h-1-j), \tag{7}$$

and

$$\mathbf{P_I}(j,i,2) = \min(i, w-1-i). \tag{8}$$

$\mathbf{I_C^Y}$, $\mathbf{I_G}$ and $\mathbf{P_I}$ are concatenated and fed into a ResNet to learn $\mathbf{M}$. The GCA weight values $\mathbf{W_{GCA}}$ share some insights with the bilateral filter [31]. $D_T(\mathbf{I_C^Y}(j,i),x_m)$ only considers the intensity differences like the range filtering within the bilateral filter, and $\mathbf{M}(j,i)$ considers the factor of positions like the domain filtering within the bilateral filter.

D. Structure similarity loss

To encourage the structure similarity between $\mathbf{I_C^Y}$ and $\mathbf{I_G}$, the structure similarity loss measures the differences between $\widehat{\mathbf{I}}_\mathbf{C}^Y$ and $\mathbf{I_G}$ with SSIM as the metric, i.e.

$$L_{structure} = 1 - SSIM(\widehat{\mathbf{I}}_\mathbf{C}^Y, \mathbf{I_G}), \tag{9}$$

E. Cycle consistency loss

To encourage the similarity of the Cb and Cr color channels between the second-time colorization result $\mathbf{R_C^{'}}$ and the reference color image $\mathbf{R_C}$, we propose the cycle consistency loss to measure the differences between the Cb and Cr color channels of $\mathbf{R_C^{'}}$ and $\mathbf{R_C}$. We use SSIM as the metric, i.e.

$$L_{cycle} = 1 - \frac{1}{2}(SSIM(\mathbf{R_C^{'Cb}}, \mathbf{R_C^{Cb}}) + SSIM(\mathbf{R_C^{'Cr}}, \mathbf{R_C^{Cr}})), \tag{10}$$

F. Spatial smoothness loss

We introduce the spatial smoothness loss to encourage spatial smoothness of the 3-D weight volume $\mathbf{V^W}$ in the WAC network so as to obtain spatial smooth colorization result. We assume that in the vertical and horizontal dimensions, neighboring pixels should have similar weights, so the loss is defined as

$$L_{smooth} = \frac{\sum\limits_{(j,i,k)} \sum\limits_{(j',i')\in\Omega(j,i)} |\mathbf{V^W}(j,i,k) - \mathbf{V^W}(j',i',k)|}{N}, \tag{11}$$

where Ω is the 4-neighboring pixels in the vertical and horizontal dimensions, and $N = \sum\limits_{(j,i,k)} \sum\limits_{(j',i')\in\Omega(j,i)} 1$.

TABLE I
SUMMARY OF OUR CYCLE CNN ARCHITECTURE. EACH 2-D OR 3-D CONVOLUTIONAL LAYER REPRESENTS A BLOCK OF CONVOLUTION, BATCH NORMALIZATION AND RELU (UNLESS OTHERWISE SPECIFIED).

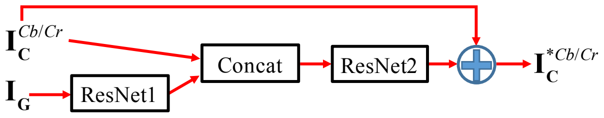
	Layer Description	**Output Tensor Dim.**
	ResNet in the WAC network	
1	5×5 conv, n feat.	$h \times w \times n$
2	3×3 conv, n feat.	$h \times w \times n$
3	3×3 conv, n feat.	$h \times w \times n$
	residue connection (add layer 1 and 3 feat.)	$h \times w \times n$
4-17	(repeat layers 2,3 and residual connection)×7	$h \times w \times n$
	ResNet in the GCA network	
1	5×5 conv, n feat.	$h \times w \times n$
2	3×3 conv, n feat.	$h \times w \times n$
3	3×3 conv, n feat.	$h \times w \times n$
	residue connection (add layer 1 and 3 feat.)	$h \times w \times n$
4-17	(repeat layers 2,3 and residual connection)×7	$h \times w \times n$
18	3×3 conv, 1 feat., Activation: Sigmoid	$h \times w$
	ResNet1 in the refinement network	
1	5×5 conv, n feat.	$h \times w \times n$
2	3×3 conv, n feat.	$h \times w \times n$
3	3×3 conv, n feat.	$h \times w \times n$
	residue connection (add layer 1 and 3 feat.)	$h \times w \times n$
4-17	(repeat layers 2,3 and residual connection)×7	$h \times w \times n$
18	3×3 conv, 1 feat.	$h \times w$
	ResNet2 in the refinement network	
1	5×5 conv, n feat.	$h \times w \times n$
2	3×3 conv, n feat.	$h \times w \times n$
3	3×3 conv, n feat.	$h \times w \times n$
	residue connection (add layer 1 and 3 feat.)	$h \times w \times n$
4-17	(repeat layers 2,3 and residual connection)×7	$h \times w \times n$
18	3×3 conv, 1 feat., no ReLu	$h \times w$
	3-D U-Net in the WAC network	
1	3-D conv, $3 \times 3 \times 3$, n feat.	$h \times w \times d \times n$
2	3-D conv, $3 \times 3 \times 3$, n feat.	$h \times w \times d \times n$
3	3-D conv, $3 \times 3 \times 3$, $2n$ feat., stride 2	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times 2n$
4	3-D conv, $3 \times 3 \times 3$, $2n$ feat.	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times 2n$
5	3-D conv, $3 \times 3 \times 3$, $2n$ feat.	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times 2n$
6-14	(repeat layer 3, 4, 5)×3	$\frac{h}{16} \times \frac{w}{16} \times \frac{d}{16} \times 2n$
15	$3 \times 3 \times 3$, 3-D trans conv, $2n$ feat., stride 2	$\frac{h}{8} \times \frac{w}{8} \times \frac{d}{8} \times 2n$
	residual connection (add layer 15 and 11)	$\frac{h}{8} \times \frac{w}{8} \times \frac{d}{8} \times 2n$
16	$3 \times 3 \times 3$, 3-D trans conv, $2n$ feat., stride 2	$\frac{h}{4} \times \frac{w}{4} \times \frac{d}{4} \times 2n$
	residual connection (add layer 16 and 8)	$\frac{h}{4} \times \frac{w}{4} \times \frac{d}{4} \times 2n$
17	$3 \times 3 \times 3$, 3-D trans conv, $2n$ feat., stride 2	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times 2n$
	residual connection (add layer 17 and 5)	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times 2n$
18	$3 \times 3 \times 3$, 3-D trans conv, n feat., stride 2	$h \times w \times d \times n$
	residual connection (add layer 18 and 2)	$h \times w \times d \times n$
19	$3 \times 3 \times 3$, 3-D trans conv, 1 feat.	$h \times w \times d$
20	Softmax	$h \times w \times d$

G. Full objective

Combining all above losses, the overall objective we aim to optimize is:

$$L = \lambda_1 L_{structure} + \lambda_2 L_{cycle} + \lambda_3 L_{smooth}, \qquad (12)$$

where λ_1, λ_2, and λ_3 control the relative importance of the corresponding terms respectively. The values are set as $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 0.1$ in this paper. With the guidance of these losses, we successfully learn the Cycle-CNN without any synthesized data for training.



$\mathbf{I}_{\mathbf{C}}^{Cb/Cr}$
$\mathbf{I}_{\mathbf{G}}$
ResNet1
Concat
ResNet2
$\mathbf{I}_{\mathbf{C}}^{*Cb/Cr}$

Fig. 7. The refinement CNN that refines the Cb and Cr channels of the colorization result $\mathbf{I}_{\mathbf{C}}$ using the input gray image $\mathbf{I}_{\mathbf{G}}$ as guidance to obtain the refined result $\mathbf{I}_{\mathbf{C}}^{*Cb/Cr}$.

H. Color refinement

The Cb and Cr color channels of the first-time colorization result $\mathbf{I}_{\mathbf{C}}$ may have errors in occlusion regions. To correct these errors, we use the input gray image $\mathbf{I}_{\mathbf{G}}$ as guidance to refine the Cb and Cr color channels of the colorization result $\mathbf{I}_{\mathbf{C}}$. We follow [2] to build the network. As shown in Fig. 7, the input gray image $\mathbf{I}_{\mathbf{G}}$ is fed into a ResNet to get its feature. The extracted feature and $\mathbf{I}_{\mathbf{C}}^{Cb/Cr}$ are concatenated and then fed into another ResNet to get the residue color map $\Phi(\mathbf{I}_{\mathbf{C}}^{Cb/Cr}, \mathbf{I}_{\mathbf{G}})$. By adding $\mathbf{I}_{\mathbf{C}}^{Cb/Cr}$ and $\Phi(\mathbf{I}_{\mathbf{C}}^{Cb/Cr}, \mathbf{I}_{\mathbf{G}})$, the refined Cb and Cr color channels, i.e. $\mathbf{I}_{\mathbf{C}}^{*Cb}$ and $\mathbf{I}_{\mathbf{C}}^{*Cr}$, are obtained. And we concatenate $\mathbf{I}_{\mathbf{G}}$, $\mathbf{I}_{\mathbf{C}}^{*Cb}$ and $\mathbf{I}_{\mathbf{C}}^{*Cr}$ to get the final result $\mathbf{I}_{\mathbf{C}}^{*}$.To train this model, we use the second-time colorization results $\mathbf{R}_{\mathbf{C}}^{'}$ as inputs and the Cb and Cr channels of the original input color image $\mathbf{R}_{\mathbf{C}}$ as ground-truth, and we use SSIM as the loss function for training.

I. Network architecture

We use ResNet as the backbone for our model. As shown in Fig. 6, 5, and 7, we use 4 ResNets in total in the WAC network, GCA network, and the refinement network. The detailed layer description is shown in Table I. They have similar structures. The first layer is with 5×5 kernel, followed by 7 residue blocks with 3×3 kernel and a residue connection. The last layers of the 4 ResNets are different for different goals. *BatchNorm* layers and *ReLu* layers are added after each convolution layer. The filter number n of the ResNet is a hyper-parameter, which is set as 16 in this paper.

We follow [32] to use the 3-D U-Net for the 3-D regulation in the WAC network. The layer description of the 3-D U-Net is also shown in Table I.

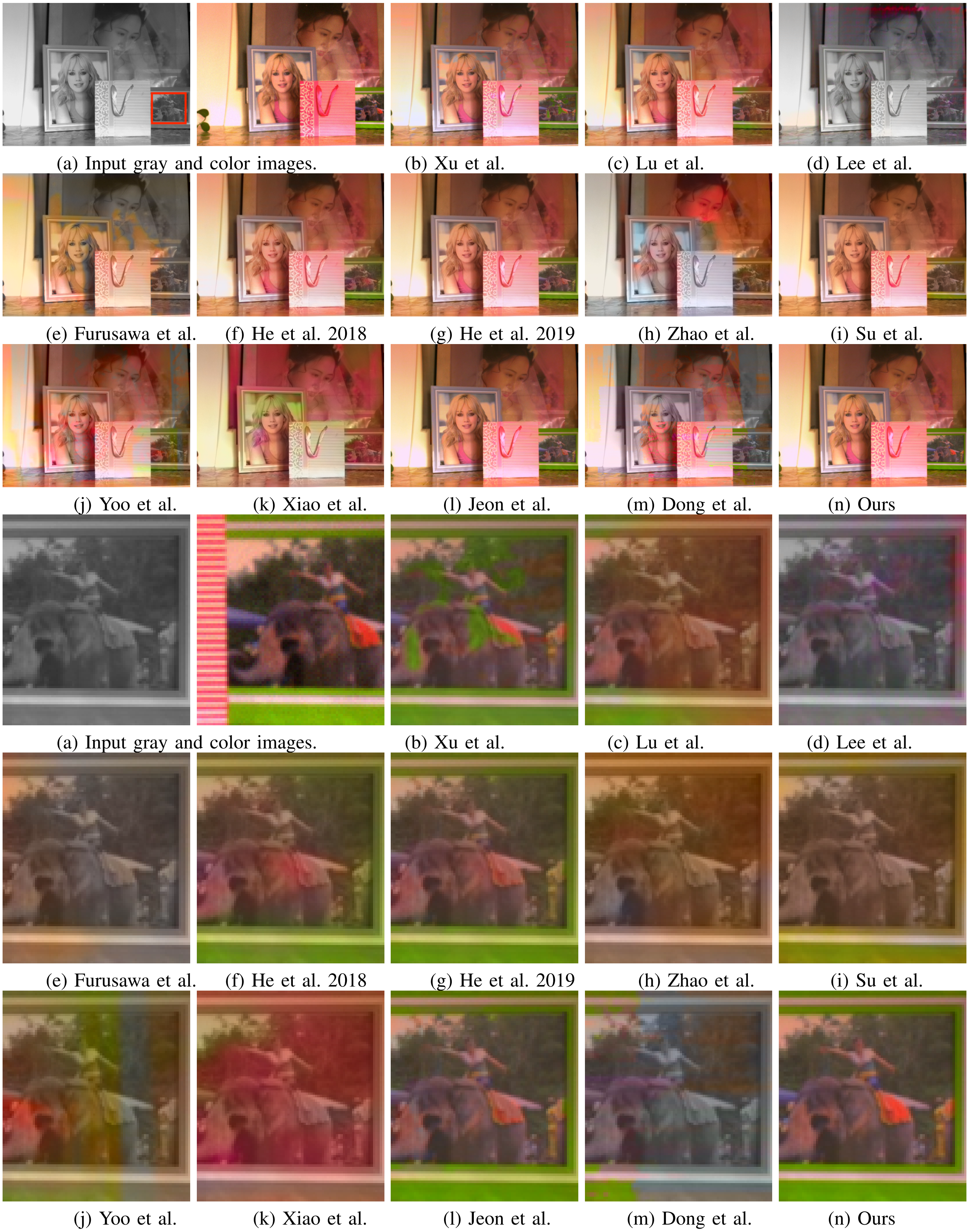
IV. EXPERIMENTAL RESULTS

A. Dataset

We use one monochrome camera and one color camera to shoot 1000 pairs of gray and color images to build the dataset, named Real Dataset. The monochrome and color cameras are rectified using the method of [34]. The cameras are the monochrome and color versions of the same camera, i.e. the MVCAM-SU1000C camera.

B. Implementation details

The proposed deep convolutional network is implemented with Tensorflow. All models are optimized with RMSProp and a constant learning rate of 0.001. We train with a batch size of 1 using a 256×512 randomly located crop from the input images. The images of the dataset is randomly divided into the training set, which contains 700 pairs of images, and the testing set, which contains 300 pairs of images. All the models are run on a server with an Intel I7 CPU and 4 NVIDIA Titan-X GPUs. The training time is about 23 hours and the testing time is shown in Table III.



(a) Input gray and color images. (b) Xu et al. (c) Lu et al. (d) Lee et al.

(e) Furusawa et al. (f) He et al. 2018 (g) He et al. 2019 (h) Zhao et al. (i) Su et al.

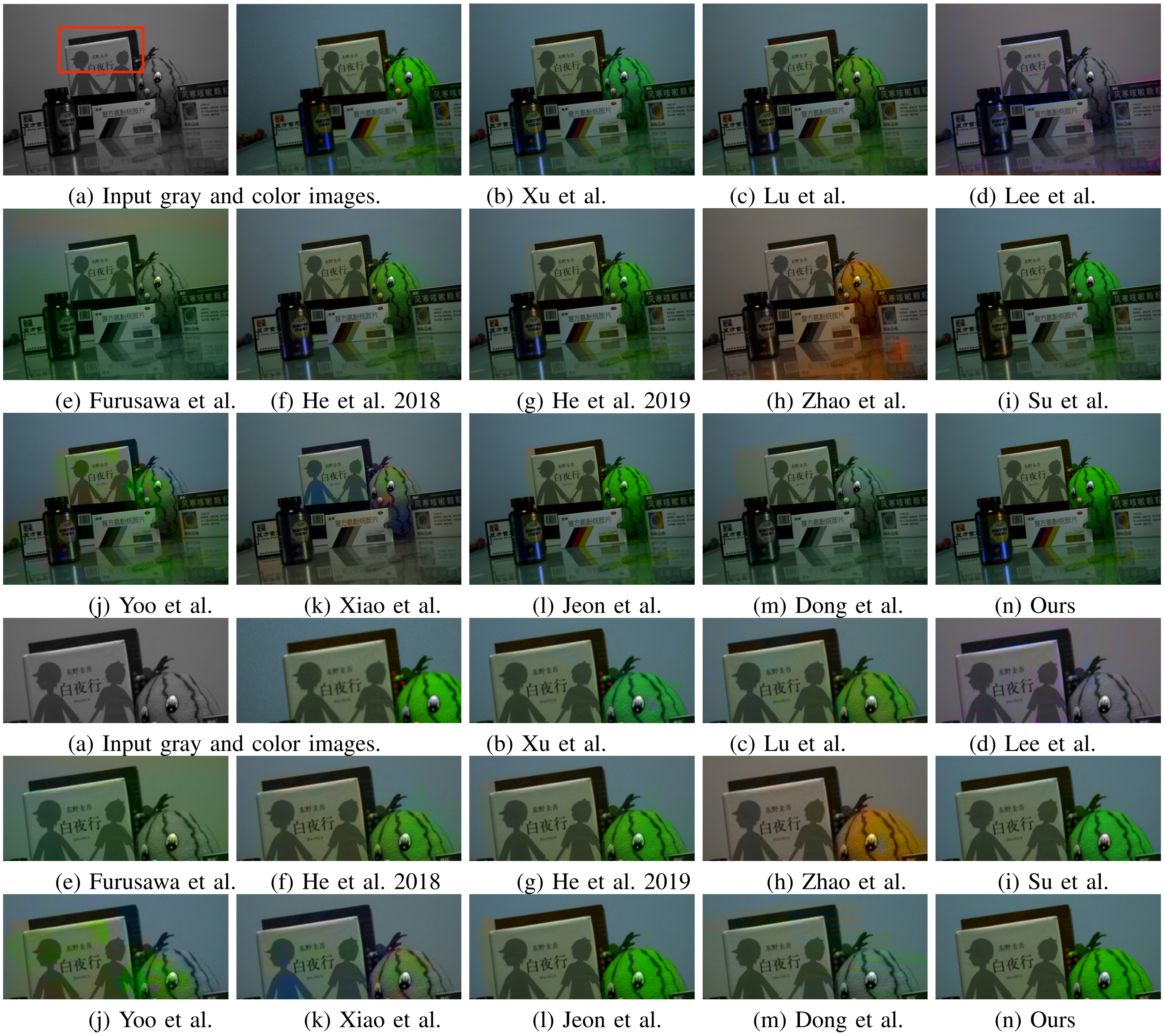
(j) Yoo et al. (k) Xiao et al. (l) Jeon et al. (m) Dong et al. (n) Ours

(a) Input gray and color images. (b) Xu et al. (c) Lu et al. (d) Lee et al.

(e) Furusawa et al. (f) He et al. 2018 (g) He et al. 2019 (h) Zhao et al. (i) Su et al.

(j) Yoo et al. (k) Xiao et al. (l) Jeon et al. (m) Dong et al. (n) Ours

Fig. 8. Examples to compare the colorization results of all the comparison methods with ours.



(a) Input gray and color images. (b) Xu et al. (c) Lu et al. (d) Lee et al.
(e) Furusawa et al. (f) He et al. 2018 (g) He et al. 2019 (h) Zhao et al. (i) Su et al.
(j) Yoo et al. (k) Xiao et al. (l) Jeon et al. (m) Dong et al. (n) Ours
(a) Input gray and color images. (b) Xu et al. (c) Lu et al. (d) Lee et al.
(e) Furusawa et al. (f) He et al. 2018 (g) He et al. 2019 (h) Zhao et al. (i) Su et al.
(j) Yoo et al. (k) Xiao et al. (l) Jeon et al. (m) Dong et al. (n) Ours

Fig. 9. Examples to compare the colorization results of all the comparison methods with ours.

C. Comparison algorithms:

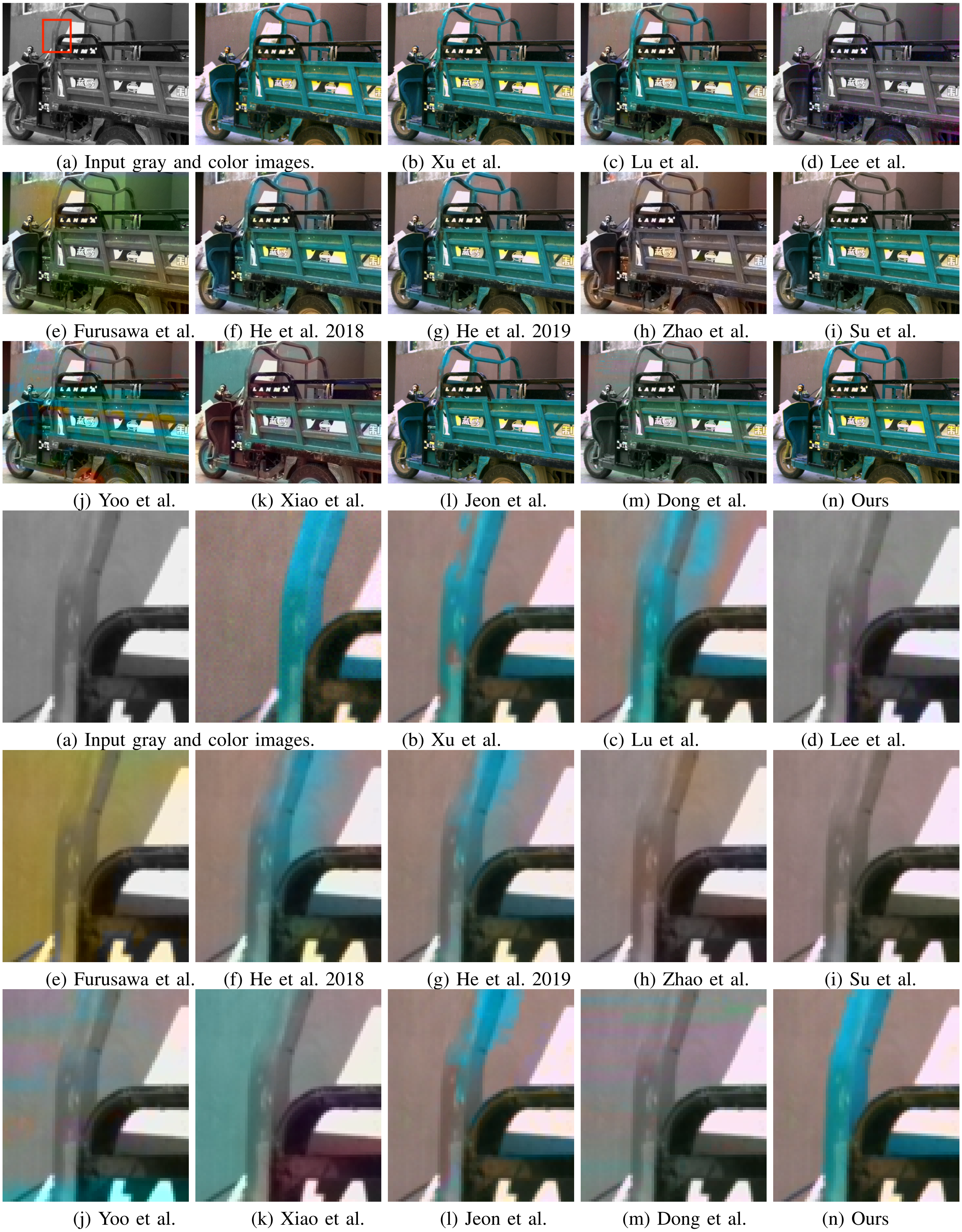
We compare with state-of-the-art reference-based colorization algorithms, i.e. the methods of Xu et al. [23], Lu et al. [24], Lee et al. [25], Furusawa et al. [19], He et al. 2018 [21], and He et al. 2019 [22], deep learning based automatic colorization algorithms, i.e. the methods of Zhao et al. [7], Su et al. [8], Yoo et al. [9] and Xiao et al. [10] and state-of-the-art monochrome-color dual-lens colorization algorithms, i.e. the methods of Jeon et al. [1] and Dong et al. [27].

For fair comparison, the automatic colorization methods are fine-tuned on the Real Dataset by using the color images as the ground-truth and their de-colored results as the input gray images. The learning based reference-based methods and monochrome-color colorization methods are all supervised methods and the Real Dataset cannot provide training data for their training or fine-tuning, because the ground-truth Cb and Cr values of the input monochrome image are not available in the Real Dataset. So we use one of the synthesized dataset, i.e. the SceneFlow Dataset, to fine-tune them in a cycle way, i.e. train them with the first-time colorization and second-time colorization alternatively in different training epochs. The non-learning based method, e.g. Jeon et al., does not need training, so we directly perform it for our problem.

D. Comparison with other colorization methods on Real Dataset

The qualitative results are shown in Figs. 8, 9 and 10. As shown, our method has better results than the comparison methods.

The colorization qualities of the state-of-the-art CNN-based automatic colorization methods, including Zhao et al. [7], Su et al. [8], Yoo et al. [9] and Xiao et al. [10], are worse than most of the reference-based methods and ours, especially in regions with colorful details and textures. It is because they are



(a) Input gray and color images. (b) Xu et al. (c) Lu et al. (d) Lee et al.

(e) Furusawa et al. (f) He et al. 2018 (g) He et al. 2019 (h) Zhao et al. (i) Su et al.

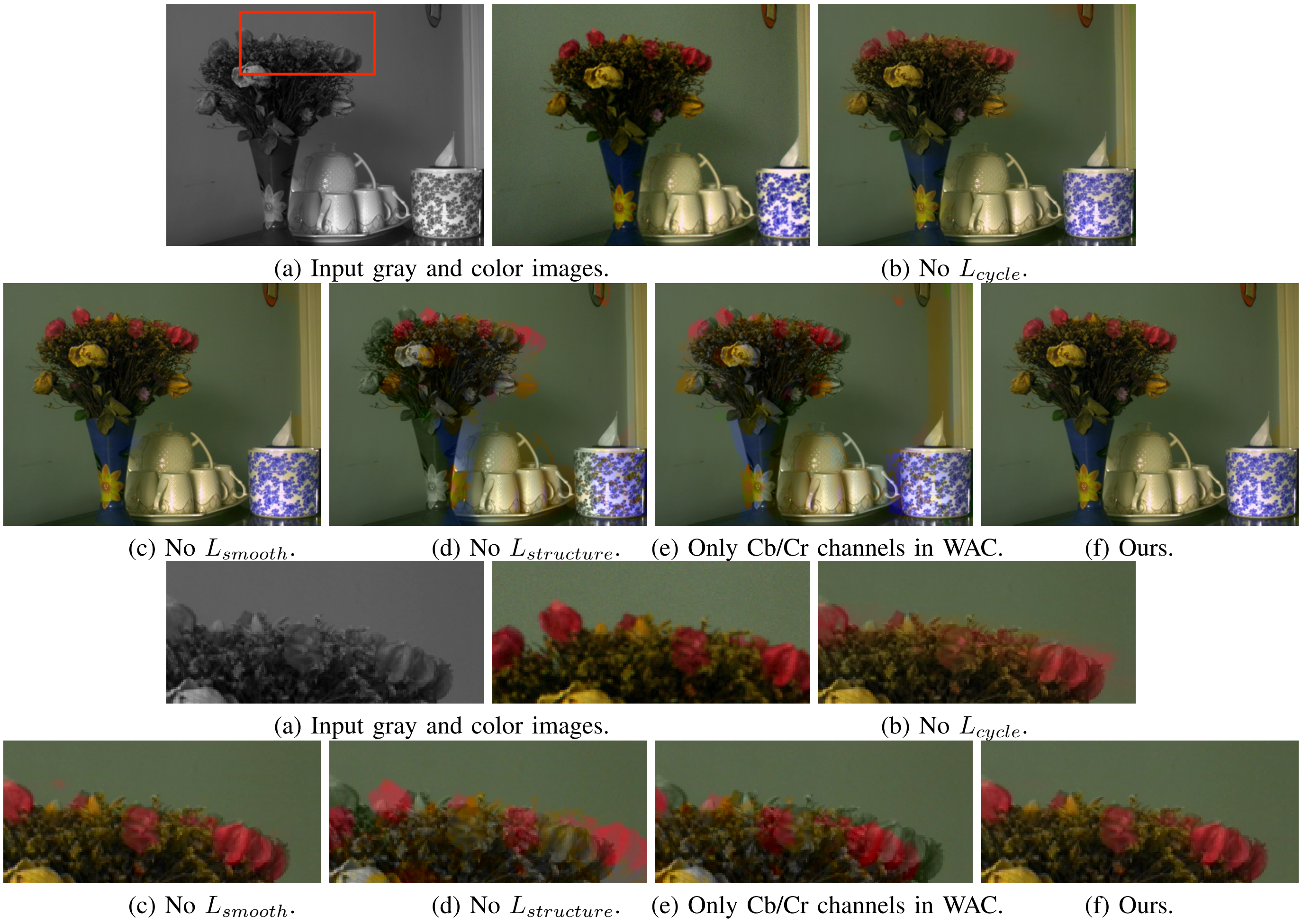
(j) Yoo et al. (k) Xiao et al. (l) Jeon et al. (m) Dong et al. (n) Ours

(a) Input gray and color images. (b) Xu et al. (c) Lu et al. (d) Lee et al.

(e) Furusawa et al. (f) He et al. 2018 (g) He et al. 2019 (h) Zhao et al. (i) Su et al.

(j) Yoo et al. (k) Xiao et al. (l) Jeon et al. (m) Dong et al. (n) Ours

Fig. 10. Examples to compare the colorization results of all the comparison methods with ours.



(a) Input gray and color images.

(b) No L_{cycle}.

(c) No L_{smooth}.

(d) No $L_{structure}$.

(e) Only Cb/Cr channels in WAC.

(f) Ours.

(a) Input gray and color images.

(b) No L_{cycle}.

(c) No L_{smooth}.

(d) No $L_{structure}$.

(e) Only Cb/Cr channels in WAC.

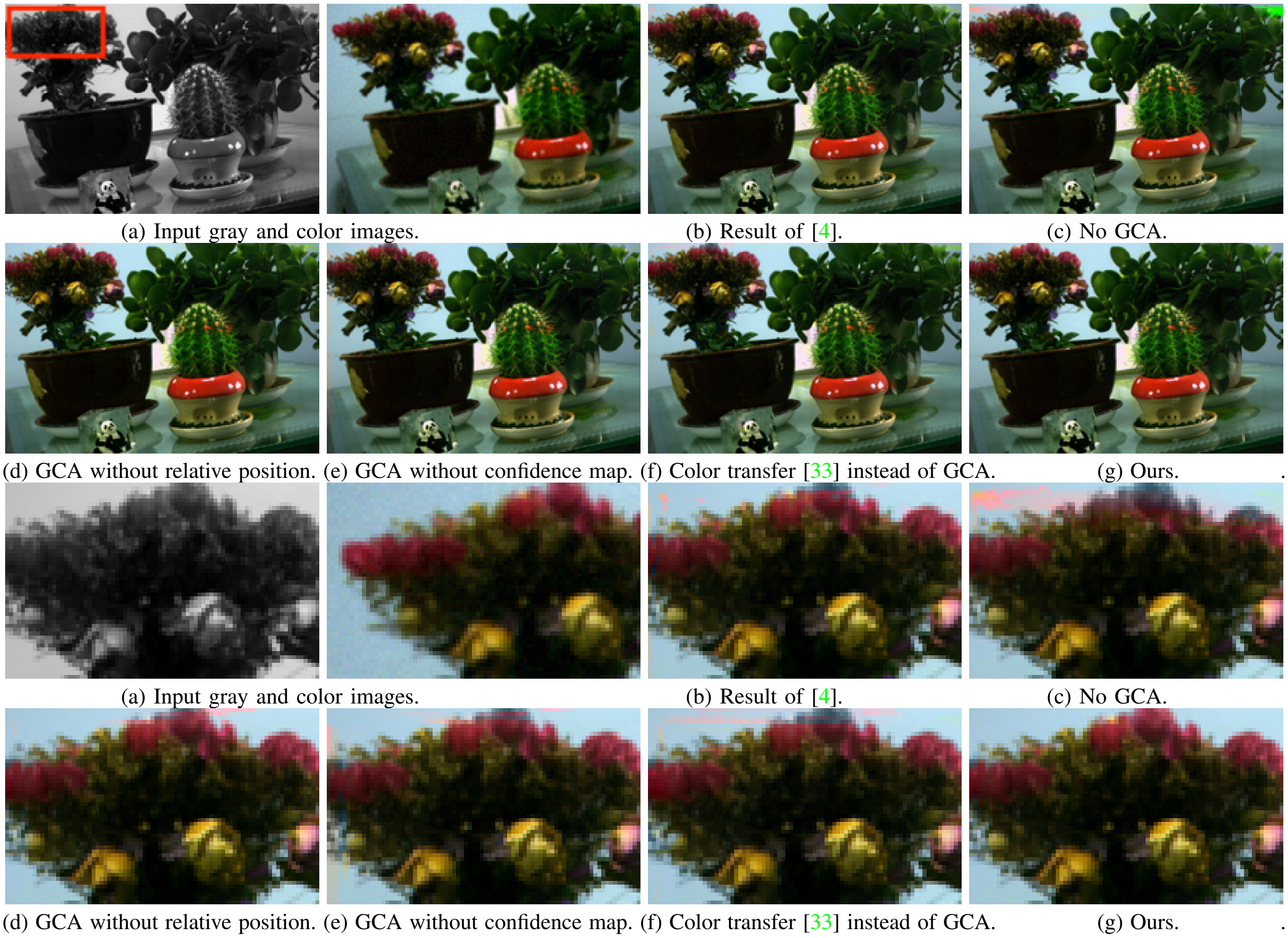
(f) Ours.

Fig. 11. Example results in the ablation study about the three losses, i.e. the structure similarity loss $L_{structure}$, the cycle consistency loss L_{cycle}, and the spatial smoothness loss L_{smooth}, and the WAC network without inferring Y channel.

solving different problems. The input in these methods is only one single gray image. The reference color image, which could provide much useful color information during the colorization, is not utilized at all.

The state-of-the-art reference-based colorization methods, i.e. Xu et al. [23], Lu et al. [24], Lee et al. [25], Furusawa et al. [19], He et al. 2018 [21], and He et al. 2019 [22], and monochrome-color dual-lens colorization algorithms, i.e. Jeon et al. [1] and Dong et al. [27], are all supervised methods and the models cannot be trained or fine-tuned on the real data from monochrome-color dual-lens systems. Because the pixels between the pair of gray and color images of the real data usually have different luminance and the distortions are complicated and blind, the differences make the colorization challenging and the comparison methods usually fail to generate good results in all regions within the images.

In detail, the method of Xu et al. [23] proposes a two-step coarse-to-fine architecture to firstly obtain a coarse result by matching basic feature statistics and secondly refine the coarse result to generate the final result, but, due to the lack of accurate confidence estimation of wrongly colorized pixels, the wrongly colorized regions of the coarse result may pollute the neighboring regions during the second step, leading to errors of the final colorization results. The method of Lu et al. [24] makes use of the prior knowledge of colors contained in the training data to fuse the semantic colors and global color distribution from the reference image to generate the final color images. However, while the basic colors are correct with the help of the prior knowledge and semantics of objects, the colors of details and textures can be hardly accurate. Lee et al. [25] propose a sketch based colorization method, but their results are poor for the real data from monochrome-color dual-lens systems. It is because the assumptions of the input images are quite different and the combination of specially-designed modules for sketch images, e.g. the outline extractor, the augmented-self reference generation, the feature transfer module, etc., is not a proper choice for our problem. The result of Furusawa et al. [19] is not good enough because the method assumes that the images are manga images but in our problem the images are general images. He et al.'s results, including [21], [22], could not achieve high accuracy. They are designed under the assumption that the pair of images are visually very different but semantically similar. Due to different assumptions from our problem, they do not consider locality and spatial smoothness of the correspondence. This causes many inconsistent correspondence matches, which will



(a) Input gray and color images. (b) Result of [4]. (c) No GCA.

(d) GCA without relative position. (e) GCA without confidence map. (f) Color transfer [33] instead of GCA. (g) Ours. .

(a) Input gray and color images. (b) Result of [4]. (c) No GCA.

(d) GCA without relative position. (e) GCA without confidence map. (f) Color transfer [33] instead of GCA. (g) Ours. .

Fig. 12. Example results in the ablation study about the GCA network.

cause wrong colorization. In addition, the perceptual loss minimizes the semantic differences of unnatural colorization. The result looks natural but is not always faithful to the ground truth colors, e.g. some small regions have different colors from neighboring regions, but they are wrongly colorized to have similar colors with neighboring regions. Jeon et al.'s method [1] has better results than the other comparison methods. Although they are designed for monochrome-color system and the features are robust to luminance variance and distortions, the hand-crafted pipelines are not competing with our deep learning based model. Dong et al. [27] have poor results because they are trained on synthesized data. Among the synthesized data, the pixels between the pair of gray and color images always have the same luminance and the distortions, e.g. blur and noise, are manually added. On the contrary, in the real data, the pixels between the pair of gray and color images usually have different luminance and the distortions are complicated and blind. Due to different characteristics of data in the synthesized and real datasets, Dong et al. have poor results for real data in some cases.

A user study is also performed. There are 30 annotators in total. The annotation choices include five score level, i.e. 'Perfect', 'Few Errors', 'Partly Wrong', 'Mostly Wrong', and 'Totally Wrong'. The annotators are asked to annotate every colorization result of the 12 comparing methods and ours. The whole set of images in the user study are 100 pairs that are randomly selected from our Real Dataset. And each annotator annotates 1300 colorization results in total. To avoid outlier annotation, we will let each annotator randomly re-annotate some results and see the annotations as outlier if the annotation differences are beyond one score level. The results

TABLE II
AVERAGE PSNR (DB) AND SSIM VALUES OF THE SECOND-TIME COLORIZATION RESULTS OF DIFFERENT COLORIZATION METHODS ON THE REAL DATASET.

	PSNR	SSIM
Xu	32.41	0.8961
Lu	33.67	0.9135
Lee	23.62	0.8225
Furusawa	25.53	0.8521
He18	33.68	0.8986
He19	34.02	0.9042
Zhao	33.28	0.9106
Su	34.12	0.9204
Yoo	35.80	0.9193
Xiao	27.99	0.8455
Jeon	34.44	0.9067
Dong	31.25	0.8741
Ours	**42.99**	**0.9707**

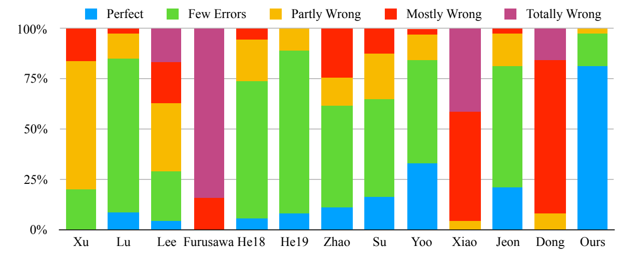


Fig. 13. User study results.

are shown in Fig. 13. This shows we can get ‘Perfect’ and ‘Few Errors’ scores in most cases and our method gets much higher perceptual scores than the others.

Quantitative evaluation is also performed by evaluating the PSNR and SSIM values between the second-time colorization results and the input color image. Due to the lack of ground-truth color information of input gray images, we cannot perform quantitative evaluation for the first-time colorization results. But, according to the cycle consistency property, the second-time colorization results can still reflect the colorization quality to some extend. So we use all the methods to do the colorization twice. The results are in Table II, which show that we outperform largely than the other methods. The processing time of different methods are shown in Table III. This shows that, while increasing the colorization accuracy largely, our method does not increase the computational complexity a lot.

E. Comparison with other colorization methods on Synthesized Dataset

Due to the importance of quantitative evaluation, we perform our method on the traditional synthesized dataset of [2], [1]. The datasets include Cityscapes [35], Middlebury [36], Sintel [37], and SceneFlow [38]. We use all the comparing methods and ours to do the colorization twice, and the quantitative results are shown in Table IV and V. As shown, on the four synthesized datasets, without using the annotated ground-truth Cb and Cr channel values of the first-time colorization results, the proposed self-supervised method achieves comparable results in comparison with the supervised method [27] which is specifically designed for the synthesized datasets and relies on the annotated ground-truth Cb and Cr channel values for training. And the proposed self-supervised method could have higher results than all the other methods. The results show the effectiveness of the combination of the GCA network and the structure similarity loss for measuring the quality of the first-time colorization result. We also test the linear correlation coefficients (LCC) between our first-time colorization results and second-time colorization results on the four datasets. The results are shown in Table VI. As shown, they have very high correlation. This verifies our insight of cycle colorization consistency and provide support that the results in Table II could reflect the colorization quality of different methods.

F. Ablation study

The ablation study compares a number of different model variants and justifies our design choices. We wish to evaluate the importance of the key ideas in this paper: 1) the designs of the losses and 2) the improvement of structure similarity loss via the GCA network in comparison with our previous conference version [4]. So, first, we try to remove each of the three losses, i.e. the cycle consistency loss, the structure similarity loss, and the spatial smoothness loss, and also let the WAC network only estimate Cb and Cr channels. Table VII shows the summary performance of different model variants. Fig. 11 shows some qualitative examples. The results show that any of these variants will degrade the colorization accuracy. In the case of only Cb and Cr channels in WAC, the first-time colorization results, which is shown in Fig. 11, are very poor, while the second-time colorization results, which is shown in Table VII, are very good. It is because the lack of estimating the Y channel makes the structure similarity loss meaningless and thus the training of the model is over-fitted to the cycle consistency loss which misleads the training and the mode collapse problem occurs. We also test it on the SceneFlow dataset. The average PSNR(dB)/SSIM values of the first-time colorization results and the second-time colorization results are 37.22/0.795 and 45.02/0.993. From the objective experimental results, we can notice the big quality differences between the first-time and second-time colorization, while our method, as shown in Tables and , has similar colorization qualities between the first-time and second-time colorization results. This verifies the contributions of all these designs of our method. Second, we remove the key component of the structure similarity loss via GCA network, including using the learned structure similarity loss in [4] instead of the GCA network, removing the whole GCA network, removing the estimation of the confidence map within the GCA network, removing the relative position map within the GCA network, and using the global color transfer method [33] to substitute for the GCA network. The summary performance of different model variants is also shown in Table VII. And Fig. 12 shows some qualitative examples. The results show that any of these variants will degrade the colorization accuracy, which verify the contributions of these key components within the GCA network.

V. Conclusion

We have presented a novel CNN model, named Cycle CNN, for colorization in real monochrome-color dual-lens system. It can be trained directly using the real data from monochrome-color camera systems without any synthesized data. The proposed method use the Weighted Average Colorization (WAC) network to do the colorization twice. In addition, we introduce the Global Curve Adjustment (GCA) network and the structure similarity loss for measuring the quality of the first-time colorization result, the cycle consistency loss for measuring the quality of the second-time colorization result, and the spatial smoothness loss to encourage spatial smoothness of the colorization result. Our method achieves superior performance than the state-of-the-art methods for colorizing real data.

TABLE III

PROCESSING TIME (MS) OF DIFFERENT METHODS FOR IMAGES WITH DIFFERENT RESOLUTIONS. THE NON-LEARNING BASED METHOD OF JEON [1] IS RUN ON CPU, AND THE DEEP LEARNING BASED METHODS ARE RUN ON GPU.

	Xu	Lu	Lee	Furusawa	He18	He19	Zhao	Su	Yoo	Xiao	Jeon	Dong	Ours
1024 × 800	931	8518	508	8149	3184	121931	11284	920	12457	65735	93662	313	803
512 × 400	304	3274	239	2361	713	34836	3141	977	3147	24087	23861	112	251
256 × 200	164	1227	131	636	154	8132	853	785	840	8395	6018	40	85

TABLE IV

AVERAGE PSNR VALUES (DB) OF THE FIRST-COLORIZATION RESULTS AND THE SECOND-COLORIZATION RESULTS OF DIFFERENT COLORIZATION METHODS ON THE SYNTHESIZED DATASETS. CT, MB, ST, AND SF ARE SHORT FOR THE DATASETS OF CITYSCAPES, MIDDLEBURY, SINTEL, AND SCENEFLOW, RESPECTIVELY.

	First-time colorization				Second-time colorization			
	CT	MB	ST	SF	CT	MB	ST	SF
Xu	38.83	33.52	37.96	33.62	39.51	36.06	38.48	33.41
Lu	39.78	29.81	38.16	39.08	39.22	29.93	38.45	38.93
Lee	30.58	23.74	27.10	29.05	29.05	23.94	27.07	28.69
Furusawa	35.21	31.13	32.43	28.62	34.42	31.13	31.73	28.13
He18	39.17	35.81	36.72	32.81	40.36	36.51	37.93	33.26
He19	39.53	36.72	36.37	32.46	40.62	36.59	37.48	33.15
Zhao	31.92	23.92	27.24	28.93	30.82	23.21	26.86	28.31
Su	31.75	23.64	29.33	35.60	29.14	23.76	28.64	27.51
Yoo	24.74	23.43	26.79	29.50	29.91	29.02	29.31	34.18
Xiao	38.10	27.55	31.08	32.15	38.12	27.66	31.09	31.98
Jeon	39.33	36.80	36.12	31.32	40.54	36.11	37.39	31.79
Dong	44.87	**42.53**	**44.46**	**45.71**	44.55	42.18	**44.05**	45.15
Ours	**45.22**	42.30	43.95	45.50	**45.26**	**42.35**	43.94	**45.67**

TABLE V

AVERAGE SSIM VALUES OF THE FIRST-TIME COLORIZATION RESULTS AND THE SECOND-TIME COLORIZATION RESULTS OF DIFFERENT METHODS ON THE FOUR SYNTHESIZED DATASETS. CT, MB, ST, AND SF ARE SHORT FOR THE DATASETS OF CITYSCAPES, MIDDLEBURY, SINTEL, AND SCENEFLOW, RESPECTIVELY

	First-time colorization				Second-time colorization			
	CT	MB	ST	SF	CT	MB	ST	SF
Xu	0.899	0.942	0.923	0.898	0.905	0.949	0.927	0.906
Lu	0.969	0.895	0.948	0.954	0.962	0.891	0.947	0.951
Lee	0.870	0.830	0.777	0.807	0.826	0.805	0.776	0.789
Furusawa	0.843	0.861	0.795	0.798	0.852	0.873	0.793	0.805
He18	0.953	0.950	0.949	0.924	0.965	0.957	0.973	0.929
He19	0.955	0.957	0.955	0.927	0.962	0.957	0.963	0.931
Zhao	0.965	0.878	0.867	0.894	0.961	0.867	0.861	0.891
Su	0.957	0.883	0.905	0.944	0.948	0.874	0.900	0.904
Yoo	0.902	0.867	0.846	0.898	0.960	0.873	0.856	0.890
Xiao	0.971	0.891	0.912	0.909	0.966	0.886	0.910	0.907
Jeon	0.953	0.958	0.943	0.927	0.953	0.950	0.959	0.928
Dong	**0.987**	**0.988**	**0.988**	**0.992**	0.986	**0.986**	**0.990**	0.989
Ours	0.983	0.983	0.984	0.989	**0.987**	0.984	0.988	**0.992**

TABLE VI

LCC BETWEEN OUR FIRST-COLORIZATION RESULTS AND OUR SECOND-COLORIZATION RESULTS ON THE FOUR SYNTHESIZED DATASETS. THE RESULTS ARE EVALUATED USING PSNR AND SSIM, RESPECTIVELY.

	CT	MB	ST	SF
PSNR	0.9930	0.9926	0.9822	0.9885
SSIM	0.9884	0.9844	0.9915	0.9917

TABLE VII

ABLATION STUDY. WE SHOW AVERAGE SSIM VALUES OF THE SECOND-TIME COLORIZATION RESULTS OF DIFFERENT VARIANTS OF OUR MODEL ON THE REAL DATASET.

	SSIM
No cycle consistency loss	0.8140
No structure similarity loss	0.9192
No spatial smoothness loss	0.9432
Previous method in [4]	0.9547
No GCA	0.9236
GCA without confidence map	0.9637
GCA without relative position map	0.9669
Color transfer [33] instead of GCA	0.9603
No Y channel within WAC	0.9799
No color refinement	0.9701
Ours	0.9707

ACKNOWLEDGMENT

This work is funded by the National Nature Science Foundation of China (No. 61802026 and 61806016) and the Fundamental Research Funds for the Central University (No. 2019RC39).

REFERENCES

[1] H. G. Jeon, J. Y. Lee, S. Im, H. Ha, and I. S. Kweon, “Stereo matching with color and monochrome cameras in low-light conditions,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4086–4094, 2016.

[2] X. Dong, W. Li, X. Wang, and Y. Wang, “Learning a deep convolutional network for colorization in monochrome-color dual-lens system,” *AAAI Conference on Artificial Intelligence*, 2019.

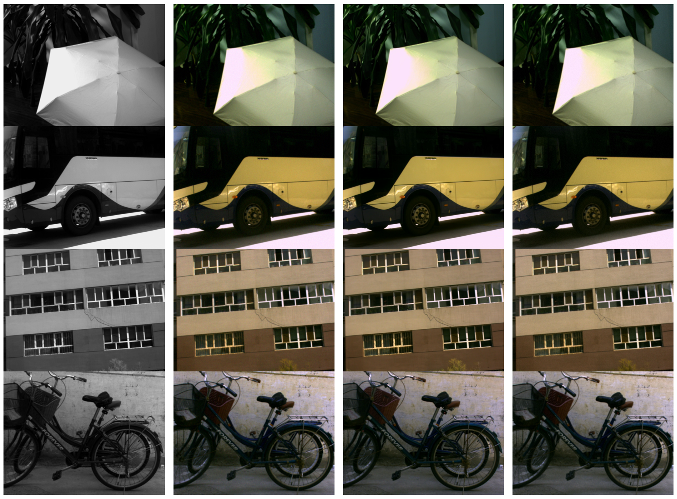
[3] P. Debevec and J. Malik, “Recovering high dynamic range radiance maps from photographs,” *SIGGRAPH*, 1997.

[4] X. Dong, W. Li, X. Wang, and Y. Wang, “Cycle-cnn for colorization towards real monochrome-color camera systems,” *AAAI Conference on Artificial Intelligence*, 2020.

[5] R. Zhang, P. Isola, and A. Efros, “Colorful image colorization,” *European Conference on Computer Vision*, pp. 649–666, 2016.

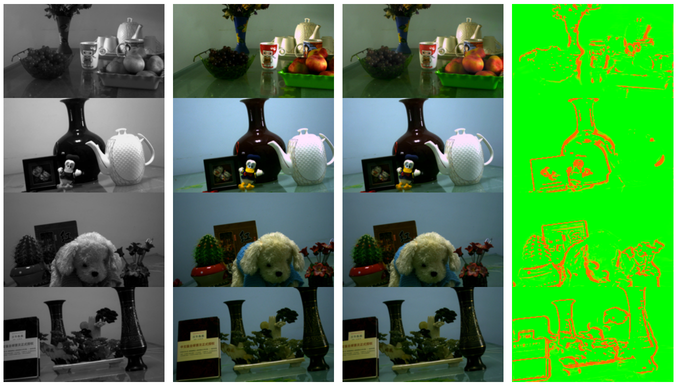
[6] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification,” *ACM Transactions on Graphics*, vol. 35, no. 4, 2016.

[7] J. Zhao, J. Han, L. Shao, and C. Snoek, “Pixelated semantic coloriza-



(a) Input gray and color images. (b) First-time result. (c) Second-time result.

Fig. 14. Examples of our first-time colorization results and second-time colorization results.



(a) Input gray and color images. (b) Our result. (c) Confidence map.

Fig. 15. Examples to visualize the learned confidence maps within the GCA module given the input pair of gray and color images. In the visualized confidence maps, green indicates high values while red indicates low values.

tion," *IJCV*, vol. 128, no. 3, pp. 818–834, 2020.

[8] J. Su, H. Chu, and J. Huang, "Instance-aware image colorization," *CVPR*, pp. 7968–7977, 2020.

[9] S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, and J. Choo, "Coloring with limited data: Few-shot colorization via memory augmented networks," *CVPR*, pp. 11 283–11 292, 2019.

[10] C. Xiao, C. Han, Z. Zhang, J. Qin, T. Wong, G. Han, and S. He, "Example-based colourization via dense encoding pyramids," *Computer Graphics Forum*, vol. 39, no. 12, pp. 1–14, 2019.

[11] H. Bahng, S. Yoo, W. Cho, D. K. Park, Z. Wu, X. Ma, and J. Choo, "Coloring with words: Guided image colorization through text-based palette generation," *ECCV*, pp. 431–447, 2018.

[12] V. Manjunatha, M. Iyyer, J. B. Graber, and L. Davis, "Learning to color from language," *North American Chapter of the Association for Computational Linguistics*, 2018.

[13] H. Kim, H. Y. Jhoo, E. Park, and S. Yoo, "Tag2pix: Line art colorization using text tag with secat and changing loss," *ICCV*, 2019.

[14] R. Zhang, J. Zhu, P. Isola, X. Geng, A. Lin, T. Yu, and A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.

[15] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM transactions on graphics*, vol. 23, no. 3, pp. 689–694, 2004.

[16] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," *ACM transactions on graphics*, vol. 21, no. 3, pp. 277–280, 2002.

[17] R. Ironi, D. Cohen-Or, and D. Lischinski, "Colorization by example," *Rendering Techniques*, pp. 201–210, 2005.

[18] R. K. Gupta, A. Y. S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong, "Image colorization using similar images," *ACM international conference on Multimedia*, pp. 369–378, 2012.

[19] C. Furusawa, K. Hiroshiba, K. Ogaki, and Y. Odagiri, "Comicolorization: semi-automatic manga colorization," *SIGGRAPH Asia*, 2017.

[20] M. He, J. Liao, L. Yuan, and P. Sander, "Neural color transfer between images," *Arxiv*, 2017.

[21] M. He, D. Chen, J. Liao, P. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM SIGGRAPH*, 2018.

[22] M. He, J. Liao, D. Chen, L. Yuan, and P. Sander, "Progressive color transfer with dense semantic correspondences," *ACM Transactions on Graphics*, vol. 38, no. 13, 2019.

[23] Z. Xu, T. Wang, F. Fang, Y. Sheng, and G. Zhang, "Stylization-based architecture for fast deep exemplar colorization," *CVPR*, pp. 9363–9372, 2020.

[24] P. Lu, J. Yu, X. Peng, Z. Zhao, and X. Wang, "Gray2colornet: Transfer more colors from reference image," *ACM MM*, pp. 3210–3218, 2020.

[25] J. Lee, E. Kim, Y. Lee, D. Kim, J. Chang, and J. Choo, "Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence," *CVPR*, pp. 5801–5810, 2020.

[26] B. Zhang, M. He, J. Liao, P. Sander, L. Yuan, A. Bermak, and D. Chen, "Deep exemplar-based video colorization," *CVPR*, 2019.

[27] X. Dong, W. Li, X. Hu, X. Wang, and Y. Wang, "A colorization framework for monochrome-color dual-lens systems using a deep convolutional network," *IEEE Transactions on Visualization and Computer Graphics, Early Access*, 2020.

[28] J. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV*, 2017.

[29] X. Wang, A. Jabri, and A. Efros, "Learning correspondence from the cycle-consistency of time," *CVPR*, 2019.

[30] L. Wang, Y. Wang, Z. Liang, Z. Lin, J. Yang, W. An, and Y. Guo, "Learning parallax attention for stereo image super-resolution," *CVPR*, 2019.

[31] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *ICCV*, 1998.

[32] K. Alex, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," *International Conference on Computer Vision*, 2017.

[33] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, 2001.

[34] G. Bradski and A. Kaehler, "Learning opencv : Computer vision with the opencv library," 2008.

[35] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

[36] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[37] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," *European Conference on Computer Vision*, pp. 611–625, 2012.

[38] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D.Cremers, A.Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4040–4048, 2016.