A Colorization Framework for Monochrome-Color Dual-Lens Systems using a Deep Convolutional Network

Xuan Dong, Weixin Li*, Xiaoyan Hu, Xiaojie Wang, Yunhong Wang

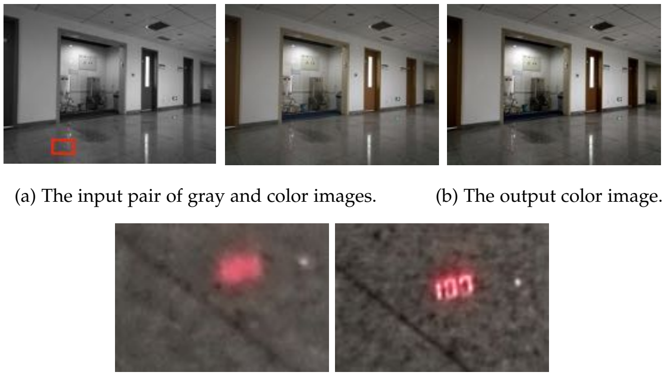
Abstract—In monochrome-color dual-lens systems, the monochrome camera can capture images with higher quality than the color camera. To obtain high quality color images, a better approach is to colorize the gray images from the monochrome camera with the color images from the color camera serving as a reference. In addition, the colorization may fail in some cases, which makes the estimation of the colorization quality a necessary step before outputting the colorization result. To solve these problems, we propose a deep convolutional network based framework. 1) In the colorization module, the proposed colorization CNN uses deep feature representations, attention operation, 3-D regulation and color correction to make use of colors of multiple pixels in the reference image for colorizing each pixel in the input gray image. 2) In the colorization quality estimation module, based on the symmetry property of colorization, we propose to utilize the colorization CNN again to colorize the gray map of the original reference color image using the first-time colorization result from the colorization module as reference. Then, the quality loss of the second-time colorization result can be used for estimating the colorization quality. Experimental results show that our method can largely outperform the state-of-the-art colorization methods and estimate the colorization quality accurately as well.

Index Terms—Colorization CNN, weight volume, color correction, colorization quality estimation.

1 Introduction

Dual-lens systems consisting of one monochrome camera and one color camera are becoming more and more popular in high-end smartphones, e.g. Huawei P9, P10, P20, etc. Between the dual lens, the monochrome one has better light efficiency than the color one [1], so the gray image from the monochrome camera has higher quality (i.e. signal-to-noise ratio) than the color image from the color camera.

To get high quality color images using the dual-lens system, we can use both cameras to shoot images at the same time and then colorize the gray image from the monochrome camera with the color image from the color camera as reference. In this way, the colorized images will have high quality in the monochrome channel and correct colors as well. An example is shown in Fig. 1. Moreover, due to occlusions, large displacement, etc., the colorization may fail to get correct colors in some cases. Thus, it is also desirable to estimate the colorization quality of each result so that our framework can judge whether the result is an outlier or an inlier, i.e. whether the colorization quality is good enough. For the inlier cases, we output the colorization results as the final results. And for the outlier cases, we output the color images from the color camera for substitution, which have lower qualities but correct colors. Thus, two problems are required to be solved in the colorization framework, i.e. 1) colorization and 2) colorization quality estimation.



(a) The input pair of gray and color images. (b) The output color image.

(c) The input and output color images in the red box region.

Fig. 1. An example of colorization in the monochrome-color dual-lens system. The input pair of images are captured by the dual-lens system of Huawei P9 phone. The output colorization result using the proposed method has high quality in the monochrome channel and correct colors.

In the literature, reference-based colorization methods, e.g. [2], [3], [1], are related to our problem. Most methods, e.g. [2], [3], usually use hand-crafted features, such as luminance, variance, etc., to search for the best-matching pixel in the reference image for each pixel in the input image. And Jeon et al. [1] use a stereo matching method, which searches for the best-matching pixel based on brightness constancy and edge similarity constraints. Although different features and matching strategies are proposed, to estimate the color of each pixel, the previous methods, e.g. [2], [3], [1], usually copy the color of only one pixel in the reference image as the result. We notice that, however, as shown in Fig. 4, for each

- *Corresponding author: Weixin Li, E-mail: weixinli@buaa.edu.cn
- Xuan Dong, Xiaoyan Hu and Xiaojie Wang are with the School of Computer Science, Beijing University of Posts and Telecommunications, China. Weixin Li and Yunhong Wang are with Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China.

Manuscript received xxxx; revised xxxx.

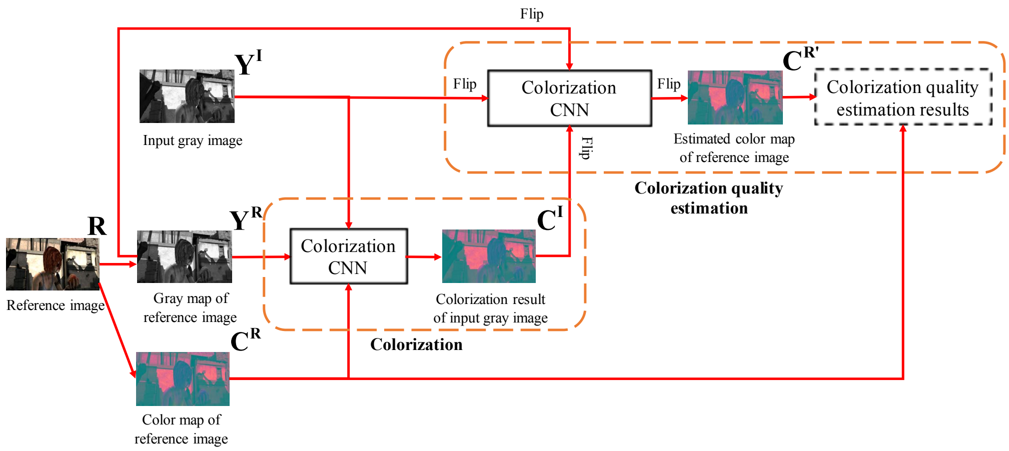

Flip
$\mathbf{Y}^{\mathbf{I}}$
Input gray image
Flip
Colorization CNN
Flip
$\mathbf{C}^{\mathbf{R}'}$
Colorization quality estimation results
Flip
Estimated color map of reference image
Colorization quality estimation
$\mathbf{R}$
Reference image
$\mathbf{Y}^{\mathbf{R}}$
Gray map of reference image
Colorization CNN
$\mathbf{C}^{\mathbf{I}}$
Colorization result of input gray image
Colorization
$\mathbf{C}^{\mathbf{R}}$
Color map of reference image

Fig. 2. Framework of the proposed method. We use the colorization CNN twice in the colorization module and the colorization quality estimation module respectively. In the colorization module, the colorization CNN generates the colorization result $\mathbf{C}^{\mathbf{I}}$. In the colorization quality estimation module, the horizontal flip operations enable the colorization CNN to be used without any changes, and the result $\mathbf{C}^{\mathbf{R}'}$ is used for estimating the colorization quality. In the training, both $\mathbf{C}^{\mathbf{I}}$ and $\mathbf{C}^{\mathbf{R}'}$ are used for training the colorization CNN so as to obtain better accuracy. (Best viewed in color)

pixel in the input image, there usually exist multiple pixels in the reference image that have the correct colors, especially in textureless and repeated texture regions. Utilizing more pixels instead of one in the reference image can help reduce noise and diminish errors in occlusion regions. Most of the existing colorization methods do not solve the colorization quality estimation problem. Traditional full-reference image quality assessment metrics, e.g. PSNR, SSIM [4], etc., cannot be directly adopted in our problem due to the lack of ground-truth color images in practice.

To deal with both issues, i.e. colorization and colorization quality estimation, in this paper, we propose a convolutional neural network (CNN) based framework. The framework is shown in Fig. 2. 1) In the colorization module, our goal is to perform the left-to-right colorization to colorize the input gray image $\mathbf{Y}^{\mathbf{I}}$ using $\mathbf{Y}^{\mathbf{R}}$ and $\mathbf{C}^{\mathbf{R}}$ of the color image $\mathbf{R}$ as reference, and we propose a convolutional neural network, named the colorization CNN, to estimate the color map $\mathbf{C}^{\mathbf{I}}$ of the input gray image $\mathbf{Y}^{\mathbf{I}}$. 2) In the colorization quality estimation module, our goal is to perform the right-to-left colorization to colorize the gray map $\mathbf{Y}^{\mathbf{R}}$ of $\mathbf{R}$ using $\mathbf{Y}^{\mathbf{I}}$ and the first-time colorization result $\mathbf{C}^{\mathbf{I}}$ as reference. We propose to make use of the colorization CNN again without any change of the network architecture for the second-time colorization. To do so, we perform horizontal flips for $\mathbf{Y}^{\mathbf{R}}$, $\mathbf{Y}^{\mathbf{I}}$ and $\mathbf{C}^{\mathbf{I}}$ before inputing them to the colorization CNN and the colorization output will be horizontally flipped again to get the second-time colorization result $\mathbf{C}^{\mathbf{R}'}$. Then, we evaluate the difference between $\mathbf{C}^{\mathbf{R}'}$ and the ground-truth color image $\mathbf{C}^{\mathbf{R}}$ to estimate the colorization quality. For the inlier cases, we will concatenate $\mathbf{Y}^{\mathbf{I}}$ and $\mathbf{C}^{\mathbf{I}}$, transfer the concatenated image into the RGB space, and output it as the final colorization result. For the outlier cases, we will output the reference image $\mathbf{R}$ as the final colorization result.

The overall structure of the colorization CNN is shown in Fig. 3. 1) Our method performs weighted average of colors of candidate pixels in the reference image to obtain the color of each pixel in the input image. To compute the weight volume that contains the weight values between all pixels in the input image and their candidate pixels in the reference image, firstly, we extract the deep features of the input gray image and the gray map of the reference image respectively by ResNet [5], and build a 4-D concatenated feature volume. Then, to obtain higher weight values between each pixel and its more useful candidate pixels for colorization, we propose an attention operation to estimate the attention weights on different candidate pixels. Next, to estimate the weight values with context information, we use the 3-D regulation to compute the 3-D weight volume. 2) After getting the weight volume, we perform the weighted average operation using the estimated weight and the reference color map to get the rough colorization result. 3) The rough colorization result may fail to have correct colors in occlusion regions, because it is possible that none of the candidate pixels in the reference image have correct colors due to occlusion. So, we propose a color correction part to correct the rough colorization result with the input gray image as guidance.

Our insight of the colorization quality estimation module is the symmetry property of colorization. In detail, the colorization quality should be similar between the left-to-right colorization, i.e. colorizing the right gray image with the left color image as reference, and the right-to-left colorization, i.e. colorizing the left gray image with the right color one as reference. Using the colorization CNN twice for the colorization module and the colorization quality estimation module can not only ensure that the colorization quality within the two modules is similar but also help train the colorization CNN to achieve higher accuracy.

Experimental results show that the proposed colorization CNN largely outperforms the state-of-the-art colorization algorithms in four datasets, including Scene Flow [6], Cityscapes [7], Middlebury [8], and Sintel [9]. The colorization quality estimation module can help estimate the colorization quality accurately, and thus the whole framework can successfully divide the colorization results into inliers and outliers.

Our contributions include: 1) For estimating the color of

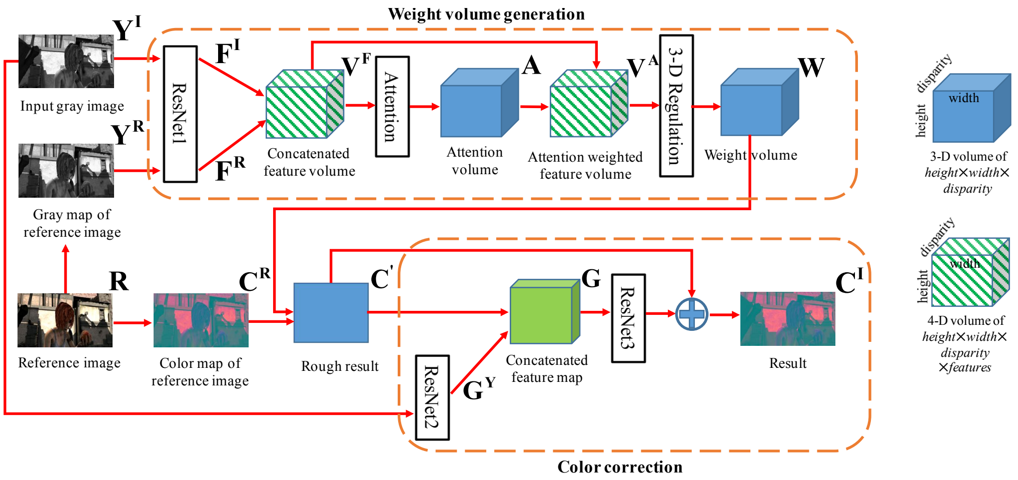


Fig. 3. The overall structure of the proposed colorization CNN. (Best viewed in color)

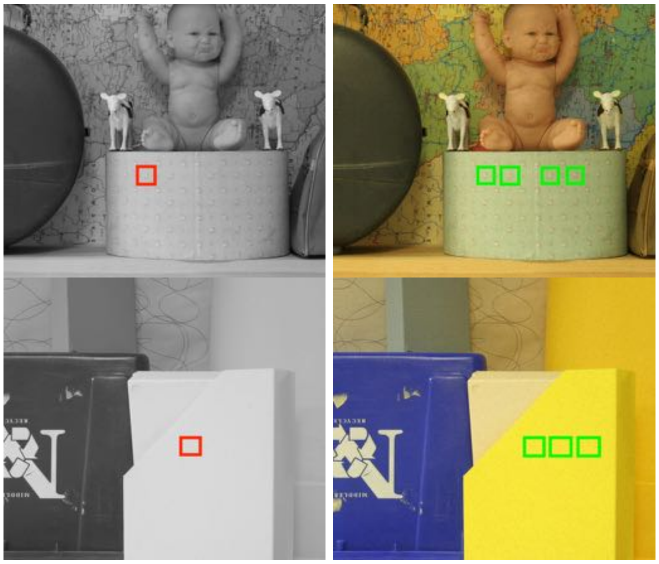


Fig. 4. Examples to show there usually exist several similar pixels (marked in green) in the reference image that could provide correct colors for a given pixel (marked in red) in the input gray image.

each pixel, we perform weighted average of colors of all candidate pixels in the reference image so as to utilize more pixels with correct colors. 2) In the proposed colorization CNN, attention mechanism and 3-D regulation are used for improving the accuracy. 3) We propose an improved color correction structure for correcting colors in occlusion regions. 4) We propose the colorization quality estimation module, which uses the colorization CNN again without any change by performing horizontal flips for the inputs and outputs.

Our framework is an extension of our previous work in [10]. Compared with [10], improvements made in this paper are as follows. 1) The previous work in [10] only proposes a colorization method, while in this paper we also propose the colorization quality estimation module. 2) The proposed framework combines both modules, i.e. the colorization module and the colorization quality estimation module, to generate high quality color images and avoid outputting outlier colorization results. 3) In the colorization CNN, we improve the color correction part so as to enhance its robustness and applicability. More details are introduced in Sec. 3 and 4.

2 Related Work

2.1 Colorization

Colorization is an important problem in computer vision and computer graphics. According to different kinds of inputs, the existing colorization methods can be divided into three categories, i.e. automatic colorization, scribble-based colorization, and reference-based colorization.

In automatic colorization, e.g. [11] and [12], the input is only a single gray image, and the methods directly colorize the gray image without any reference. Due to the lack of the reference color image, the methods usually need to learn high-level features to generate reasonable colors for objects within the image. However, the generated colors may be quite different from the ground truth, e.g. a blue ball may be colorized to be a red ball. And using these methods in our problem is not proper because the methods fail to make use of the reference color image, which provides much useful color information.

In scribble-based colorization, e.g. [13] and [14], the input includes a gray image and some color scribbles provided by the users. The scribbles are used as guidance for colorizing the gray image. However, these methods are not suitable for our problem as well, because the scribbles from the users are not available in the camera system.

Reference-based colorization algorithms, e.g. [1], [2], [3], [15], [16], [17], [18], are related to our problem. Welsh et al. [15] assume that the pixels with the same luminance value will have the same color, so they use the luminance value as the feature to search for best-matching pixels between the image pair. But, for dual-lens systems, pixels with the same luminance value have different colors in many cases. The failure of the assumption will lead to wrong colorization results. Ironi et al. [2] propose to firstly colorize the searched sparse matching pixels and then use a confidence computation method to mark low confidence pixels whose colors are propagated by neighboring colorized pixels with high confidence. However, when using the method in the dual-lens system, many non-occlusion pixels are marked as low confidence pixels, and thus the colorized pixels with high confidence are insufficient to be used for propagating the color to low confidence pixels, especially in regions of edges and small objects. Gupta et al. [3] extract features of super-pixels to search for correspondences by feature matching and use space voting for spatial consistency. But, the feature of each super-pixel is the averaged value of features of all pixels in the super-pixel. This will decrease the accuracy of the matching for our problem, leading to wrongly colorized results, especially when the image contains small objects/parts with complicated textures. Furusawa et al. [16] propose a reference-based colorization algorithm for colorizing manga images. The assumption for manga images is not always correct for general images. Thus, their results are not always good enough for solving our problem.

Jeon et al. [1] solve the same problem with ours, i.e. colorization in the monochrome-color dual-lens system. They propose a stereo matching based method to estimate the dense correspondences between the pixels of the pair of images. But, 1) the disparity estimation needs a lot of spatial smoothing and filtering, which is costly in computation and not a must-do for our problem. In addition, 2) the method in [1] corrects the colors of occlusion regions by applying spatial consistency of neighboring pixels over the whole image. But, because they do not judge the regions of occlusions where many pixels may have wrong colorization, the spatial consistency operation will pollute the neighboring correctly colorized pixels instead of correcting the errors in some cases. 3) And the confidence term in the spatial consistency operation is computed based on the super-pixel segmentation results. However, the super-pixels may be wrongly segmented, and pixels of different objects may be included in the same super-pixel. This will lead to errors after the spatial consistency operation.

Recently, some deep learning based methods are proposed, e.g. He et al. [18] and Dong et al. [19]. He et al. [18] propose a general reference-based colorization algorithm. They do not assume the pair of images are shot by the dual-lens system. Due to different assumptions from our problem, they do not consider locality and spatial smoothness and their loss minimizes the semantic differences of unnatural colorization. The result looks natural but is not always faithful to the ground truth colors. Dong et al. [19] propose to train ResNet features of pixels and use the features to search for best-matching pixels between the image pair. However, the method of searching for best-matching pixel will still fail to find correct correspondence in some regions, especially in occlusion regions. In our colorization CNN, for each pixel of the input image, we estimate the weight values of its candidate pixels in the reference image and perform weighted average of colors of all its candidate pixels to get the estimated color. In this way, we make use of multiple pixels in the reference image for colorizing each pixel, and this can help reduce noise and errors.

The monochrome-color dual-lens system is very similar with the stereo system. Another possible solution is to first use a pure stereo matching method, e.g. [20], to estimate the disparity between the images, and then copy the colors of the corresponding pixels in the reference image to the current pixels in the gray image. But, even if the estimated disparity is exactly correct, this solution can hardly generate correct results in occlusion regions, because, for those occluded pixels, their corresponding pixels in the reference image are occluded and thus cannot provide correct colors for reference.

Besides colorization, there exist some other enhancement problems in the multiple-camera system, like video retargeting [21], super resolution [22], [23], deblur [24], style transfer [25] [26], flow estimation [27], and inpainting [28]. But, these methods cannot be directly used for our problem.

2.2 Colorization quality estimation

Most existing colorization methods do not solve the colorization quality estimation problem, and they will always output the colorization results no matter they are outliers or inliers.

Traditional full-reference image quality assessment metrics, e.g. PSNR, SSIM [4], etc., need the ground-truth image at hand as the reference. They can hardly be used in our case. Although in the training stage, they can be used to evaluate the average accuracy of each colorization method because the ground-truth color images are available, in practice, our framework needs to estimate the colorization quality of each result and judge whether it belongs to the outliers without the ground-truth color images at hand.

There exist some no-reference image quality assessment metrics, e.g. BRISQUE [29], BLISS [30], etc. But they are designed to evaluate the signal noise ratio of the images, e.g. distortions caused by blur, noise, and compression. Due to different goals, the no-reference metrics are not proper for our problem either.

3 Colorization CNN

The pipeline of the colorization CNN is shown in Fig. 3. First, we generate the weight volume, which contains the weight values between each pixel in the input image and its candidate pixels in the reference image. And the weight volume is then used for performing the weighted average operation to obtain the rough colorization result. Second, in the color correction part, we jointly learn to correct the wrongly colorized pixels of the rough result using the input gray image as guidance.

The goal of the proposed weighted average operation is to utilize more useful pixels in the reference image for colorizing each pixel. The challenges are that 1) in the weighted average operation, if the weight values of the candidate pixels with incorrect colors are large, noise or even errors will be introduced to the colorization results. We thus propose an attention operation to reduce the noise/errors. The attention mechanism has been successfully used in various problems, e.g. text classification [31], and visual question answering [32]. It could help the network focus more on useful information for improving the prediction accuracy. We adopt the attention mechanism to pay more attentions on those useful candidate pixels in the reference image. This will help obtain higher weight values of the useful candidate pixels and reduce noise/errors in the colorization results. 2)

In addition, the weight volume is estimated based on the deep features of the input images. However, the features are not perfect all the time, so, the 3-D regulation [20], which learns with context information, is performed to generate the weight volume.

Colorization by the weighted average operation may fail to have correct colors in occlusion regions, because it is possible that none of the candidate pixels in the reference image have correct colors due to occlusion. To correct wrongly colorized pixels, we propose the color correction part in our network. We share similar insights with [14] that neighboring pixels with similar gray intensities should have similar colors, and the input gray image $\mathbf{Y^I}$ could provide guidance of spatial color consistency. Our method is based on the deep joint filter [33]. Our difference from [33] is that 1) we use ResNet [5] instead of traditional 2-D convolution due to good performances of ResNet in related problems. 2) And we learn the residue between the ground truth color image and the rough colorization result, because learning the residue map has proven to be more effective in related works, e.g. single image super resolution [34]. Last, 3) we use two ResNet sub-nets instead of three ResNet sub-nets [10] to build the network, which could improve the robustness and applicability of the model. Our previous work in [10] extracts features of both the rough result $\mathbf{C}'$ and $\mathbf{Y^I}$ and concatenates them. In this paper, we extract the feature of $\mathbf{Y^I}$, i.e. $\mathbf{G^Y}$, and concatenate the rough result $\mathbf{C}'$ with $\mathbf{G^Y}$. The reason is that our goal is to learn the color residue between $\mathbf{C}'$ and the ground-truth colors, and transferring $\mathbf{C}'$ to a ResNet feature, which in done in [10], is not a must-do step. By directly using the rough result $\mathbf{C}'$ for the concatenation, the robustness and applicability of the network is improved.

3.1 Formulation

Given the color image $\mathbf{R} \in \mathbb{R}^{h\times w\times 3}$ from the color camera as reference, we want to predict the color map $\mathbf{C^I}$ of the input gray image $\mathbf{Y^I} \in \mathbb{R}^{h\times w}$ from the monochrome camera. We use the *YCbCr* color space in this paper. The Y channel map of $\mathbf{R}$ is denoted as $\mathbf{Y^R}$. The *Cb* and *Cr* channel maps are denoted as $\mathbf{C^R}$. All parameters of the deep network are shared for predicting the *Cb* and *Cr* channel maps.

First, for each pixel (j, i), we propose to estimate the rough colorization result $\mathbf{C}^{'}_{j,i}$ by the weighted average of colors of its candidate pixels in the reference image, i.e.

$$\mathbf{C}^{'}_{j,i} = \sum_{k=0}^{d-1} \mathbf{W}_{j,i,k}\mathbf{C}^{\mathbf{R}}_{j,i+k}. \tag{1}$$

The range 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 is the maximum disparity. It is because the dual-lens of smart 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. $\mathbf{W}_{j,i,k}$ 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 $\mathbf{W} \in \mathbb{R}^{h\times w\times d}$ contains the weight values of all pixels and their candidate pixels.

Second, we use the input gray image $\mathbf{Y^I}$ as guidance to correct the rough result $\mathbf{C}^{'}$ by

$$\mathbf{C^I} = \mathbf{C}' + \Phi(\mathbf{C}', \mathbf{Y^I}), \tag{2}$$

where Φ denotes the operation of learning the color residue in the color correction part.

3.2 Weight volume generation

The weight volume $\mathbf{W} \in \mathbb{R}^{h\times w\times d}$ is estimated using the weight volume generation module, as shown in Fig. 3. The inputs include the input gray image $\mathbf{Y^I}$ and the gray map of the reference image $\mathbf{Y^R}$.

First, we extract the deep features $\mathbf{F^I} \in \mathbb{R}^{h\times w\times n}$ and $\mathbf{F^R} \in \mathbb{R}^{h\times w\times n}$ of $\mathbf{Y^I}$ and $\mathbf{Y^R}$ respectively by a ResNet, named ResNet1 in this paper. The hyper-parameter n is the filter number.

Then, for each pixel (j, i), we concatenate its features $\mathbf{F^I}_{j,i}$ with features of each candidate pixel $\mathbf{F^R}_{j,i}$. And the concatenated features of all pixels and their candidate pixels form the 4-D feature volume $\mathbf{V^F} \in \mathbb{R}^{h\times w\times d\times 2n}$, where

$$\mathbf{V^F}_{j,i,k} = Concat(\mathbf{F^I}_{j,i}, \mathbf{F^R}_{j,i}). \tag{3}$$

Next, the attention operation, which is achieved by using two 3-D convolution layers, is performed to obtain the attention volume $\mathbf{A}$ from the feature volume $\mathbf{V^F}$. Each element of $\mathbf{A}$, i.e. $\mathbf{A}_{j,i,k}$, is the attention weight between features of pixel (j, i) and its candidate pixel $(j, i + k)$. The attention volume $\mathbf{A}$ is used to refine the feature volume $\mathbf{V^F}$ by

$$\mathbf{V^A}_{j,i,k,p} = \begin{cases} \mathbf{V^F}_{j,i,k,p}, & p = 0 : n - 1 \\ \mathbf{A}_{j,i,k}\mathbf{V^F}_{j,i,k,p}, & p = n : 2n - 1 \end{cases} \tag{4}$$

Next, the 3-D regulation, which is proposed by [20] to learn with context, is performed to estimate the weight volume $\mathbf{W}$ from the attention weighted feature volume $\mathbf{V^A}$.

Once $\mathbf{W}$ is obtained, the rough colorization result can be obtained by Eq. 1.

3.3 Color correction

Our goal is to use the input gray image $\mathbf{Y^I}$ as guidance to correct the rough colorization result $\mathbf{C}'$, which may contain wrongly colorized pixels due to occlusions.

As shown in Fig. 3, the input gray image $\mathbf{Y^I}$ is fed into a ResNet, named ResNet2, to get its feature $\mathbf{G^Y}$. Then, the rough colorization result $\mathbf{C}'$ and $\mathbf{G^Y}$ are concatenated to form the feature map $\mathbf{G}$, which is fed into another ResNet, named ResNet3, to get the residue color map $\Phi(\mathbf{C}', \mathbf{Y^I})$. By adding $\mathbf{C}'$ and the residue color map $\Phi(\mathbf{C}', \mathbf{Y^I})$, the final colorization result $\mathbf{C^I}$ is obtained. The joint learning of the color residue can be seen as a high dimension joint filter.

3.4 Network architecture

We show our network architecture in Fig. 3. The detailed layer information is shown in Table 1.

1) In the weight volume generation module, ResNet1 has 18 convolution layers in total. The first layer is with 5×5 kernel and stride 2. Here, we downsample the data with stride 2 to reduce memory cost. The resolution is recovered

TABLE 1
Summary of the architecture of the colorization CNN. Each 2-D or 3-D convolutional layer represents a block of convolution, batch normalization and ReLu.

	Layer Description	**Output Tensor Dim.**
	Input gray image $\mathbf{Y}$	$h \times w$
	Gray map of reference Image $\mathbf{Y^R}$	$h \times w$
	ResNet1	
1	5×5 conv, n feat., stride 2	$\frac{h}{2} \times \frac{w}{2} \times n$
2	3×3 conv, n feat.	$\frac{h}{2} \times \frac{w}{2} \times n$
3	3×3 conv, n feat.	$\frac{h}{2} \times \frac{w}{2} \times n$
	add layer 1 and 3 feat. (residue connection)	$\frac{h}{2} \times \frac{w}{2} \times n$
4-17	(repeat layers 2,3 and residual connection)×7	$\frac{h}{2} \times \frac{w}{2} \times n$
18	3×3 conv, n feat., no ReLu/BN	$\frac{h}{2} \times \frac{w}{2} \times n$
	Attention	
19	3-D conv,$1 \times 1 \times 1$,n feat.,Sigmoid,no BN/ReLu	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times n$
20	3-D conv,$1 \times 1 \times 1$,1 feat.,Sigmoid,no BN/ReLu	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2}$
	3-D regulation	
21	3-D conv, $3 \times 3 \times 3$, n feat.	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times n$
22	3-D conv, $3 \times 3 \times 3$, n feat.	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times n$
23	3-D conv, $3 \times 3 \times 3$, $2n$ feat., stride 2	$\frac{h}{4} \times \frac{w}{4} \times \frac{d}{4} \times 2n$
24	3-D conv, $3 \times 3 \times 3$, $2n$ feat.	$\frac{h}{4} \times \frac{w}{4} \times \frac{d}{4} \times 2n$
25	3-D conv, $3 \times 3 \times 3$, $2n$ feat.	$\frac{h}{4} \times \frac{w}{4} \times \frac{d}{4} \times 2n$
26-34	(repeat layer 23, 24, 25)×3	$\frac{h}{32} \times \frac{w}{32} \times \frac{d}{32} \times 2n$
35	$3 \times 3 \times 3$, 3-D trans conv, $2n$ feat., stride 2	$\frac{h}{16} \times \frac{w}{16} \times \frac{d}{16} \times 2n$
	add layer 35 and 31 (residual connection)	$\frac{h}{16} \times \frac{w}{16} \times \frac{d}{16} \times 2n$
36	$3 \times 3 \times 3$, 3-D trans conv, $2n$ feat., stride 2	$\frac{h}{8} \times \frac{w}{8} \times \frac{d}{8} \times 2n$
	add layer 36 and 28 (residual connection)	$\frac{h}{8} \times \frac{w}{8} \times \frac{d}{8} \times 2n$
37	$3 \times 3 \times 3$, 3-D trans conv, $2n$ feat., stride 2	$\frac{h}{4} \times \frac{w}{4} \times \frac{d}{4} \times 2n$
	add layer 37 and 25 (residual connection)	$\frac{h}{4} \times \frac{w}{4} \times \frac{d}{4} \times 2n$
38	$3 \times 3 \times 3$, 3-D trans conv, n feat., stride 2	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times n$
	add layer 38 and 22 (residual connection)	$\frac{h}{2} \times \frac{w}{2} \times \frac{d}{2} \times n$
39	$3 \times 3 \times 3$, 3-D trans conv, 1 feat., no ReLu/BN	$h \times w \times d$
	ResNet2	
40	5×5 conv, n feat.	$h \times w \times n$
41-57	repeat layers 2-18	$h \times w \times n$
	ResNet3	
58-74	repeat layers 40-56	$h \times w \times n$
75	3×3 conv, 1 feat. (no ReLu, BN)	$h \times w$

in the last layer of the 3-D regulation. The following 16 layers are 8 repeated residue blocks and each residue block consists of 2 convolution layers with 3×3 kernel and a residue connection. *BatchNorm* layers and *ReLu* layers are added after each of the 17 convolution layers. The 18th layer is a convolution layer with 3×3 kernel and no *BatchNorm* layer or *ReLu* layer is added. The filter number n of the 18 layers of ResNet1 is a hyper-parameter, which is set as 32 in this paper. The attention operation consists of two 3-D convolution layers, i.e. layer 19 and 20 in Table 1. The kernel of both layers is $1 \times 1 \times 1$. The filter numbers are n and 1, respectively. *Sigmoid* layer is added after layer 19 and 20. The *Sigmoid* layer ensures that the attention weight ranges from 0 to 1. In the 3-D regulation operation, deep encoder-decoder designs are used, i.e. we encode sub-sampled feature maps, followed by up-sampling in a decoder. We form the 3-D regulation network with four levels of sub-sampling. For each encoder level, we apply two $3 \times 3 \times 3$ convolutions. To up-sample the volume in the decoder, we employ a 3-D transposed convolution. In addition, we add each higher resolution feature map before up-sampling. Readers may refer to [20] for more details. 2) In the color correction part, we use two ResNets, named ResNet2 and ResNet3. They have similar network structure with ResNet1. The difference between ResNet2 and ResNet1 is that in the first layer of ResNet2, the stride is set as 1 instead of 2. The difference between ResNet3 and ResNet2 is that in the last layer the filter number is 1 and no *BatchNorm* layer or *ReLu* layer is added. The parameters of ResNet2 and ResNet3 are trained separately.

TABLE 2
Two setups of the colorization benchmark. We simulate the monochrome-color dual-lens system by adding different distortions to the color image and the monochrome image. In the setup1, we add signal dependent Gaussian noise with different standard deviations, where κ represents the noise-free signal intensity [35]. In the setup2, to simulate different resolutions of the monochrome and color images, we down-sample the color image with the ratio of 0.5 using Bicubic interpolation before adding the noise. In the enhancement step, we resize the reference color image to the original resolution using Bicubic interpolation before performing the colorization.

noise std.	color camera	monochrome camera
Setup1	$0.03\sqrt{\kappa}$	$0.01\sqrt{\kappa}$
Setup2	downsample+$0.07\sqrt{\kappa}$	$0.01\sqrt{\kappa}$

4 Colorization Quality Estimation

The colorization qualities of the proposed colorization module using the colorization CNN vary a lot for different inputs. Most of the results are perfect, while some others have wrongly colorized pixels due to occlusions, large displacement, etc.

A practical solution is to automatically estimate the colorization quality of each result and then adaptively judge whether the result is an outlier or an inlier, i.e. whether the colorization quality is good enough. For the inlier cases, the colorization result can be output as the final result. For the outlier cases, we could use the color image from the color camera as the final result for substitution, which has lower quality but correct colors.

It is challenging for estimating the colorization quality of the colorization result due to the lack of ground-truth color map in practice.

Our insight for solving this problem is the symmetry property of colorization, i.e. the colorization quality should be similar between the left-to-right colorization and right-to-left colorization. So, after we perform the left-to-right colorization by the colorization CNN in the colorization module and get the colorization result $\mathbf{C^I}$ of the input image $\mathbf{I}$, in the colorization quality estimation module, we propose to perform the right-to-left colorization to colorize the gray map of the reference image $\mathbf{Y^R}$ using the image $\mathbf{I}$ as reference, where $\mathbf{I} = \{\mathbf{Y^I}, \mathbf{C^I}\}$. In the second-time colorization, we perform horizontal flips for the inputs and outputs so that the colorization CNN can be used again without any change. In this way, the color map of the reference image $\mathbf{C^R}$ can be used as the ground-truth to estimate the colorization quality of the second-time colorization result $\mathbf{C^{R'}}$, and we use the quality of $\mathbf{C^{R'}}$ to estimate the quality of the colorization. The pipeline is shown in Fig. 2.

TABLE 3
The Pearson Linear Correlation Coefficients (LCC) between the qualities of the colorization results of the colorization module and the colorization quality estimation module on the four datasets. The results are evaluated using PSNR and SSIM, respectively. CT, MB, ST, and SF are short for the datasets of Cityscapes, Middlebury, Sintel, and SceneFlow, respectively.

	CT	MB	ST	SF
PSNR	0.9130	0.9426	0.9422	0.9285
SSIM	0.9284	0.9444	0.9515	0.9317

We also verify the effectiveness of the colorization quality estimation module by testing the correlation between the

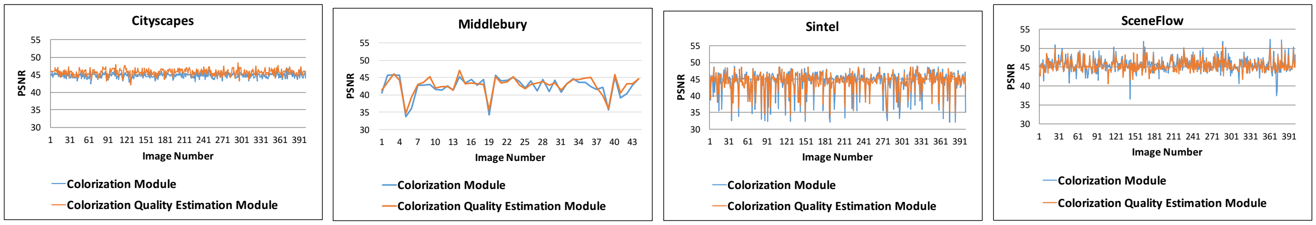


Fig. 5. PSNR values (dB) of results of the colorization module and the colorization quality estimation module in the four datasets, i.e. Cityscapes [7], Middlebury [8], Sintel [9], and SceneFlow [6]. We randomly select 400 images from Cityscapes, Sintel, and SceneFlow due to the large number of images.

results of the colorization module, i.e. $\mathbf{C^I}$, and the results of the colorization quality estimation module, i.e. $\mathbf{C^{R'}}$, among the training datasets. The PSNR values of randomly selected images in different datasets are shown in Fig. 5. The Pearson Correlation Coefficients between the qualities of $\mathbf{C^I}$ and $\mathbf{C^{R'}}$ are shown in Table 3. The statistics show that the qualities of $\mathbf{C^I}$ and $\mathbf{C^{R'}}$ are strongly correlated and provide strong support for the proposed colorization quality estimation method.

Thus, in our method, we set a threshold $T = 42$ dB in our paper, for the PSNR value of $\mathbf{C^{R'}}$. If the PSNR value of $\mathbf{C^{R'}}$ is lower than T, we see the corresponding colorization result $\mathbf{C^I}$ of the colorization module as an outlier. Otherwise, the colorization result $\mathbf{C^I}$ is treated as an inlier. And for the outlier cases, we use the original color images $\mathbf{R}$ from the color camera as the output color image. We set $T = 42$ because images with 42 dB or higher can usually be seen as high quality images.

5 Experiments

5.1 Datasets

We use four popular stereo datasets in our experiments, i.e. Cityscapes [7], Middlebury [8], Sintel [9], and SceneFlow [6]. The image pairs in these datasets are captured by dual color lens at the same time. For realistic simulations, within each image pair, we follow [1] to de-color one image. The decolored result is used as the input gray image, and the other color image is used as the input color image. In addition, we imitate the light-efficiency differences between the color and monochrome cameras by adding different distortions to the input gray images and the reference color images. We configure two different setups for this experiment. The details are shown in Table 2.

5.2 Implementation details

The proposed deep convolutional network is implemented with TensorFlow. We train our entire model in an end-to-end way from a random initialization with 15 epochs. All models are optimized with RMSProp [36] 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. We train the network on the dataset of Scene Flow, which contains 35,454 training and 4,370 testing images, on an Intel I7 and an NVIDIA Titan-X GPU. In the training, the loss function L is defined as

$$L = MSE(\mathbf{C^I}, \mathbf{C^{I*}}) + MSE(\mathbf{C^{R'}}, \mathbf{C^R}), \tag{5}$$

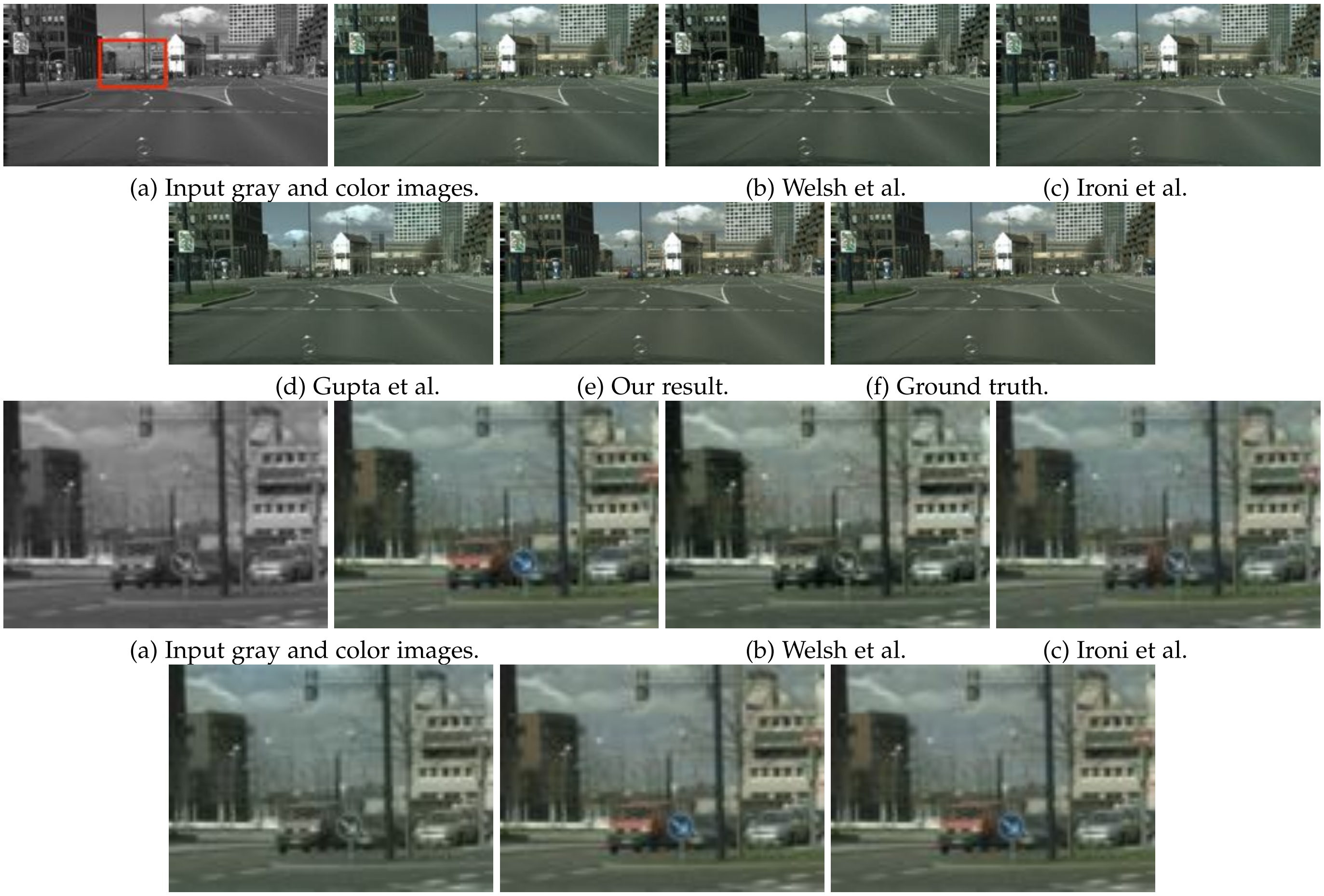
where the mean squared error (MSE) is used for measuring the quality of the prediction results of the colorization module $\mathbf{C^I}$ and the colorization quality evaluation module $\mathbf{C^{R'}}$ based on the corresponding ground-truth color maps, i.e. $\mathbf{C^{I*}}$ and $\mathbf{C^R}$. When testing the performance on the other three datasets, we directly use the model trained on Scene Flow for cross-validation.

The colorization results may have wrong colors at the image boundaries. At image boundaries, for pixels of the input gray image, the corresponding pixels of the reference image do not exist, or the appearances of the corresponding pixels are quite different. Both reasons result in the failure colorization in some cases. A practical solution we use is to cut off 5% boundary regions of the colorization results.

5.3 Experiment I: Comparison with other colorization methods

Comparison algorithms: First, we compare with the state-of-the-art reference-based colorization algorithms, i.e. the methods of Welsh et al. [15], Ironi et al. [2], Gupta et al. [3], Jeon et al. [1], Furusawa et al. [16], He et al. [18] and Dong et al. [19]. In addition, we compare with two state-of-the-art deep learning based automatic colorization algorithms, i.e. the methods of Zhang et al. [11] and Iizuka et al. [12], which could automatically colorize monochrome images without any reference images. The methods of Welsh et al. [15], Ironi et al. [2], and Gupta et al. [3] do not assume short-baseline between the pair of images. So, for each pixel in the monochrome image, the search region is the whole reference image. For fair comparison, we re-implement the methods and make the search range the same as our method, i.e. the candidate pixels are with the same vertical position and their horizontal positions range from i to $i+d-1$ as defined in Sec. 3.1. The method of Furusawa et al. is designed for colorizing manga images while we aim at general images. When performing the method of Furusawa et al., the panel is set as the whole reference image.

Results: We show the quantitative results in Tables 4 and 5. As shown, our method largely outperforms the comparison methods. And some qualitative colorization results are shown in Figs. 6, 7, and 8. As shown in Fig. 6, Welsh et al.'s method does not have good performance, because their assumption, i.e. pixels with the same grayscale intensity will have the same color value, is not true for many images. So, some regions are wrongly colorized. Ironi et al.'s method has problems at edges and small objects because many unoccluded pixels are wrongly marked as occluded pixels,



(a) Input gray and color images. (b) Welsh et al. (c) Ironi et al.

(d) Gupta et al. (e) Our result. (f) Ground truth.

(a) Input gray and color images. (b) Welsh et al. (c) Ironi et al.

(d) Gupta et al. (e) Our result. (f) Ground truth.

Fig. 6. Examples to compare the colorization results of Welsh et al. [15], Ironi et al. [2], Gupta et al. [3], and our colorization method. The region marked with the red box is shown in the bottom two rows. As shown, the comparison methods fail to recover correct colors in the marked region. This example is under Setup1 in Table 2.

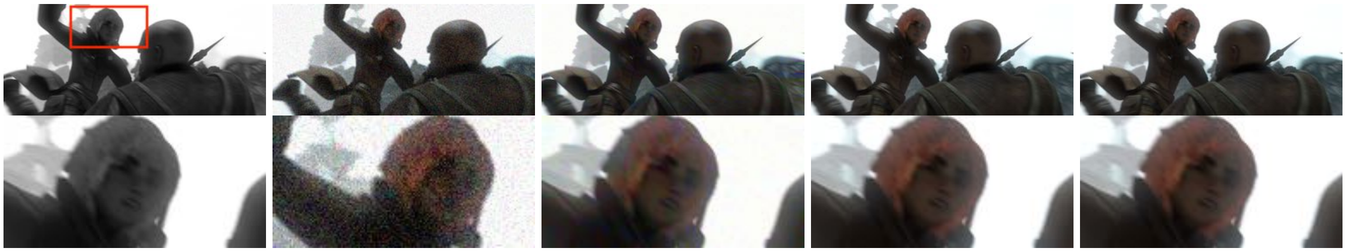
TABLE 4
Average PSNR values (dB) of different colorization methods in four datasets under Setup 1 and 2 in Table 2. CT, MB, ST, and SF are short for the datasets of Cityscapes, Middlebury, Sintel, and SceneFlow, respectively.

	PSNR(dB) under Setup1				PSNR (dB) under Setup2			
	CT	MB	ST	SF	CT	MB	ST	SF
Welsh	37.89	30.28	34.94	30.12	35.02	29.63	32.64	29.81
Ironi	38.45	32.98	36.06	31.24	35.53	30.37	32.45	30.51
Gupta	38.09	31.04	35.45	29.65	34.51	30.74	33.27	29.70
Jeon	39.33	36.80	36.12	31.32	35.24	34.68	33.89	31.71
Furusawa	34.74	30.86	32.13	28.44	32.91	29.52	32.01	27.07
He	39.05	35.63	36.28	32.15	36.13	33.38	33.17	31.26
Zhang	29.38	29.12	29.34	17.26	29.57	28.41	29.44	18.56
Iizuka	31.30	29.19	33.97	21.02	31.39	28.42	34.02	23.13
Dong	40.86	38.12	39.26	41.35	40.03	36.72	38.47	40.02
Ours	**44.87**	**42.53**	**44.46**	**45.71**	**43.65**	**40.66**	**43.12**	**44.47**

TABLE 5
Average SSIM values of different colorization methods in four datasets under Setup 1 and 2 in Table 2.

	SSIM under Setup1				SSIM under Setup2			
	CT	MB	ST	SF	CT	MB	ST	SF
Welsh	0.897	0.906	0.795	0.813	0.849	0.877	0.756	0.767
Ironi	0.897	0.940	0.918	0.890	0.778	0.714	0.812	0.743
Gupta	0.948	0.896	0.933	0.869	0.906	0.893	0.905	0.750
Jeon	0.953	0.958	0.943	0.927	0.911	0.950	0.922	0.900
Furusawa	0.841	0.860	0.794	0.795	0.825	0.782	0.728	0.734
He	0.951	0.949	0.948	0.919	0.928	0.947	0.931	0.889
Zhang	0.460	0.746	0.687	0.279	0.455	0.752	0.688	0.303
Iizuka	0.757	0.677	0.852	0.411	0.751	0.688	0.852	0.414
Dong	0.977	0.980	0.971	0.982	0.960	0.964	0.961	0.965
Ours	**0.987**	**0.988**	**0.988**	**0.992**	**0.983**	**0.980**	**0.980**	**0.987**

and thus the colorized pixels of unoccluded pixels are not enough for color propagation. Gupta et al.'s method does not perform well, especially for objects with complicated textures. It is because the features of each superpixel are obtained by averaging the feature values of all pixels in the superpixel, which will decrease the accuracy of correspondence searching for our problem. Jeon et al.'s method has better results than the other comparison methods. But they do not deal with the occlusion regions well. As shown in Fig. 7, there are occlusions between the girl and the rock behind her, and the results of their method are not correct. Furusawa et al.'s result, as shown in Fig. 8, 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, as shown in Fig. 9, could not achieve high PSNR/SSIM values because they do not consider locality and spatial smoothness of the correspondence. This causes many inconsistent correspondence matches, which will cause wrong colorization. In addition,



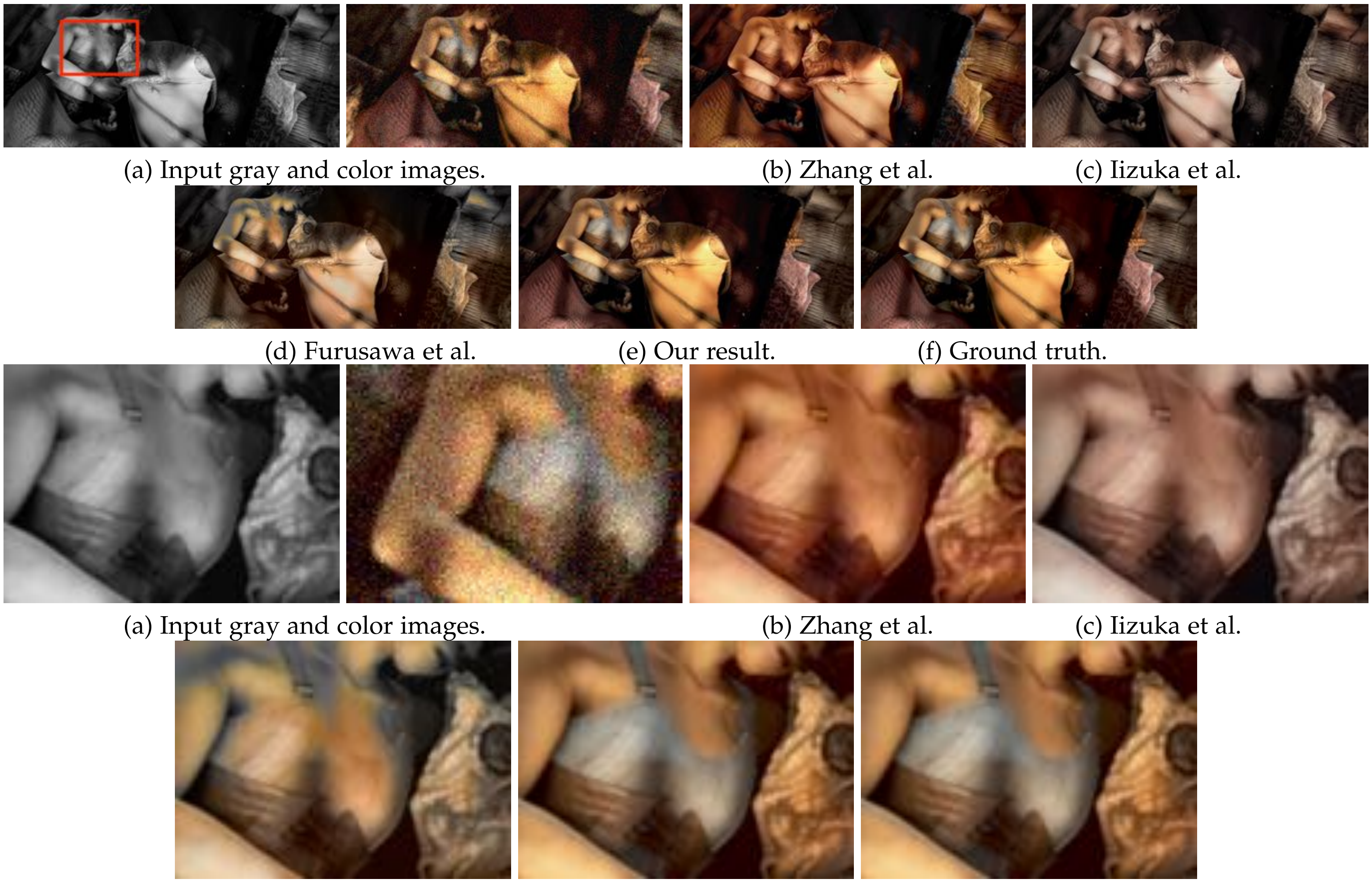
(a) The input pair of gray and color images.

(b) Result of Jeon et al.

(c) Our result.

(d) Ground truth.

Fig. 7. Examples to compare the colorization results of Jeon et al. [1] and our colorization method. The region marked with the red box is shown in the second row. As shown, Jeon et al.'s method fails to recover correct colors in the marked region. This example is under Setup2 in Table 2.



(a) Input gray and color images.

(b) Zhang et al.

(c) Iizuka et al.

(d) Furusawa et al.

(e) Our result.

(f) Ground truth.

(a) Input gray and color images.

(b) Zhang et al.

(c) Iizuka et al.

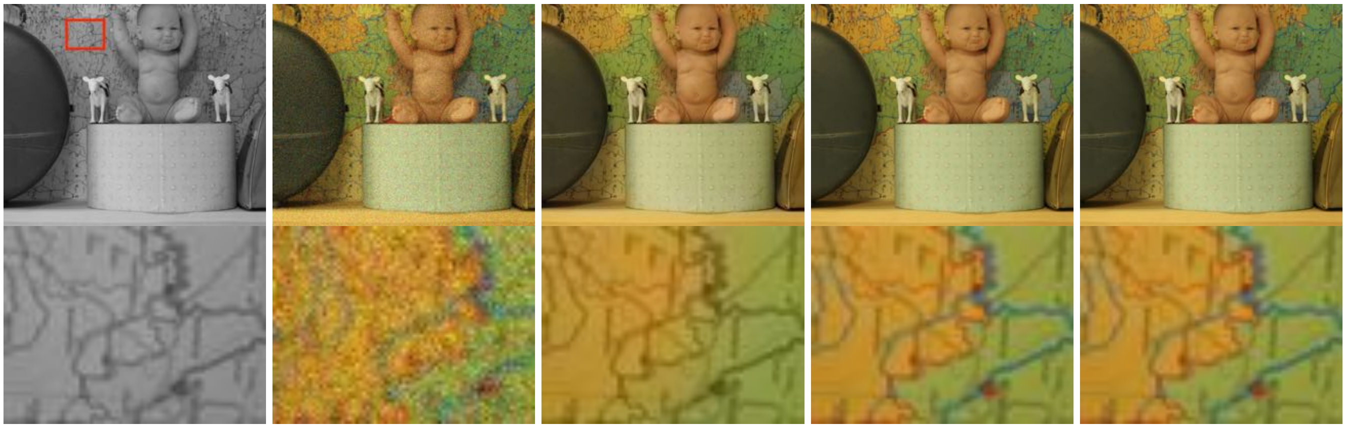
(d) Furusawa et al.

(e) Our result.

(f) Ground truth.

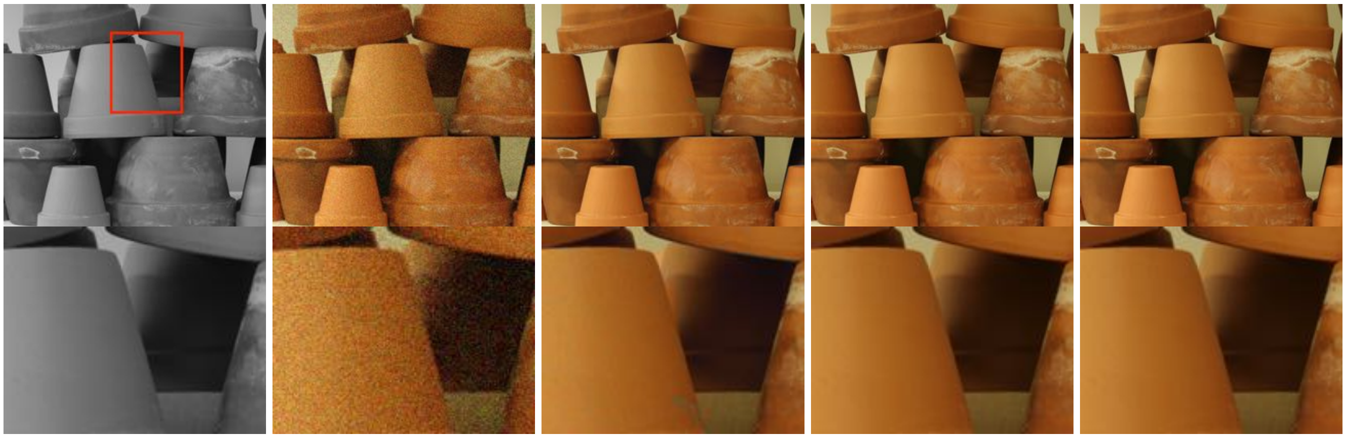
Fig. 8. Examples to compare deep learning based automatic colorization algorithms, i.e. Zhang et al. [11] and Iizuka et al. [12], manga image colorization algorithm, i.e. Furusawa et al. [16], and our algorithm. As shown, due to not using the reference images as guidance, the recovered colors of Zhang et al. and Iizuka et al. are not correct in most regions. The method of Furusawa et al. fails in most regions too, because the assumption of manga images is not true for general real-world images. The examples are from Setup2 in Table 2.

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. The results of Dong et al. have lower PSNR/SSIM values than ours because they share similar pipeline with the traditional colorization methods, i.e. extracting feature of each pixel and using the feature to search for the best-matching pixel. This pipeline fails to make use of multiple pixels with correct colors in the reference image. In addition, the proposed 3-D regulation module in the colorization CNN could improve the weight estimation with the help of spatially neighboring pixels, while Dong et al. fail to make use of spatial consistency. An example is shown in Fig. 10. The colorization qualities of the state-of-the-art CNN-based automatic colorization methods [11], [12] are worse than most of the reference-based mathods and ours. As shown in Fig. 8, their results have wrong colors in most regions. It is because they are 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.



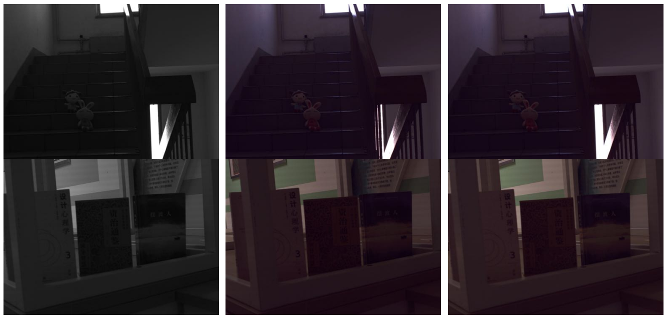
(a) The input pair of grayscale and color images. (b) Result of He et al. (c) Our result. (d) Ground truth.

Fig. 9. Examples to compare the colorization results of He et al. [18] and our method.



(a) The input pair of grayscale and color images. (b) Result of Dong et al. (c) Our result. (d) Ground truth.

Fig. 10. Examples to compare the colorization results of Dong et al. [19] and our colorization method. The marked region with red box is shown in the following rows. As shown, Dong et al. fail to recover correct colors at the marked region while our results have correct colors.



(a) The input pair of grayscale and color images. (b) Our result.

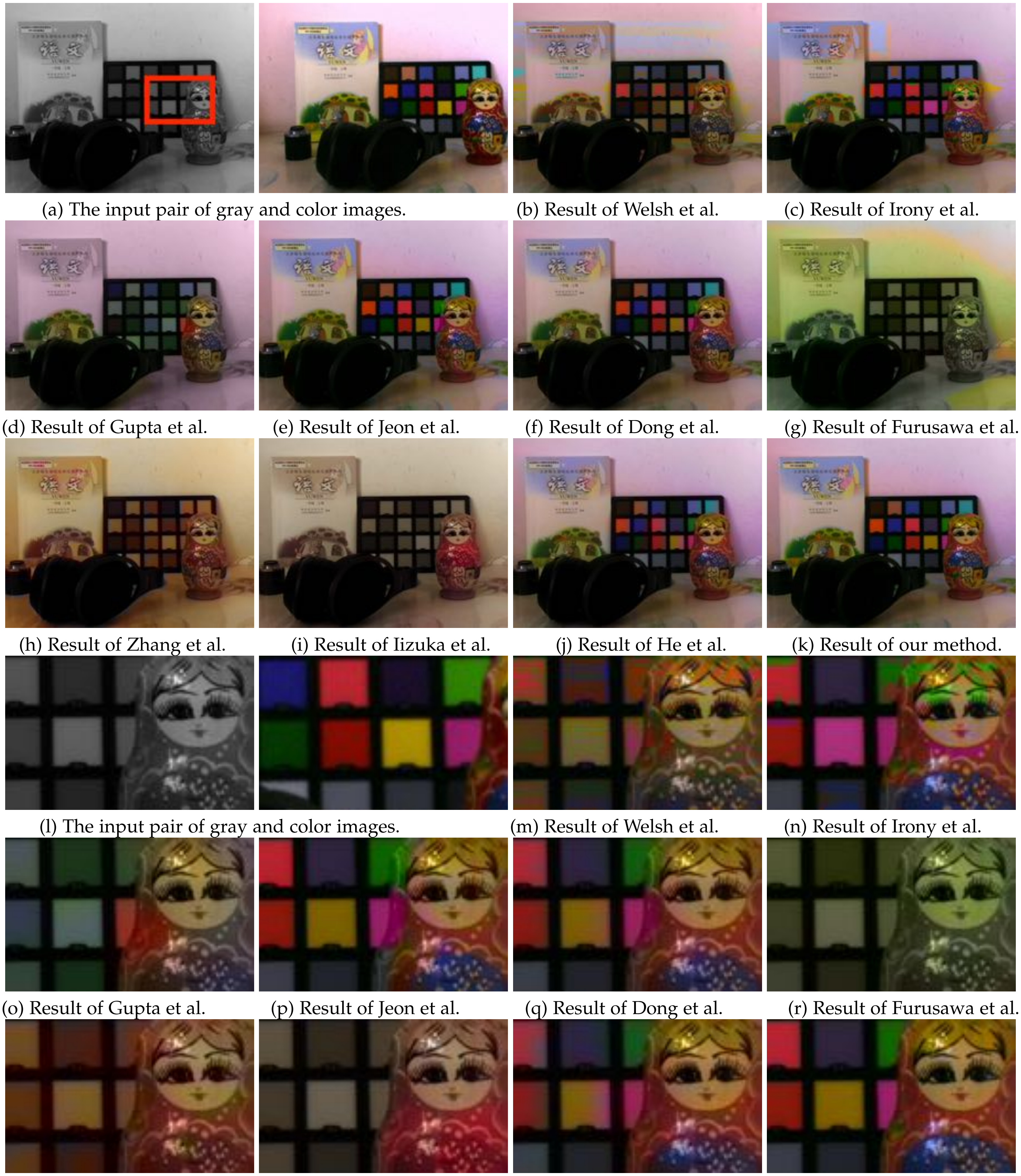
Fig. 11. Examples of our colorization method for real monochrome-color dual-lens images in low light conditions.

TABLE 6
Ablation study.

	PSNR(dB)				SSIM			
	CT	MB	ST	SF	CT	MB	ST	SF
Stereo matching model	22.17	24.51	21.72	25.31	0.755	0.697	0.700	0.763
No weighted average	39.84	38.37	40.14	40.82	0.965	0.962	0.975	0.979
No attention	41.91	40.92	41.85	42.02	0.975	0.975	0.979	0.979
No color correction	36.04	37.55	36.05	35.98	0.954	0.955	0.957	0.959
Dong et al. [10]	44.26	41.94	43.88	45.18	0.982	0.981	0.983	0.988
Simple average	16.51	18.38	13.63	18.42	0.641	0.644	0.519	0.640
Ground-truth disparity	35.48	33.82	34.24	34.84	0.875	0.808	0.823	0.848
Only colorization module	44.74	42.35	44.32	45.58	0.986	0.987	0.987	0.990
Ours	**44.87**	**42.53**	**44.46**	**45.71**	**0.987**	**0.988**	**0.988**	**0.992**

The average processing time of the comparison methods and our method for images with different resolutions is shown in Table 7. Among the processing parts of our method, the 3D U-Net for the 3-D regulation is the most computationally complex processing part, costing 76.4% of the overall processing time. The color correction part costs 10.3% of the overall processing time, and the ResNet deep feature extraction and attention parts costs 13.3% of the overall processing time. So, the acceleration should focus on optimizing the 3D U-Net part in the future. To the best of our knowledge, there are some standard acceleration solutions. Possible solutions include 1) network pruning methods like [37], 2) coarse-to-fine processing like [38], and 3) in our current implementation, we only use one GPU. In the future, multi-GPUs could be used for acceleration. In short, our work provides a good start point and baseline for the research and industry communities.

We also show some qualitative colorization results for the input images captured by the real monochrome-color



(a) The input pair of gray and color images. (b) Result of Welsh et al. (c) Result of Irony et al.

(d) Result of Gupta et al. (e) Result of Jeon et al. (f) Result of Dong et al. (g) Result of Furusawa et al.

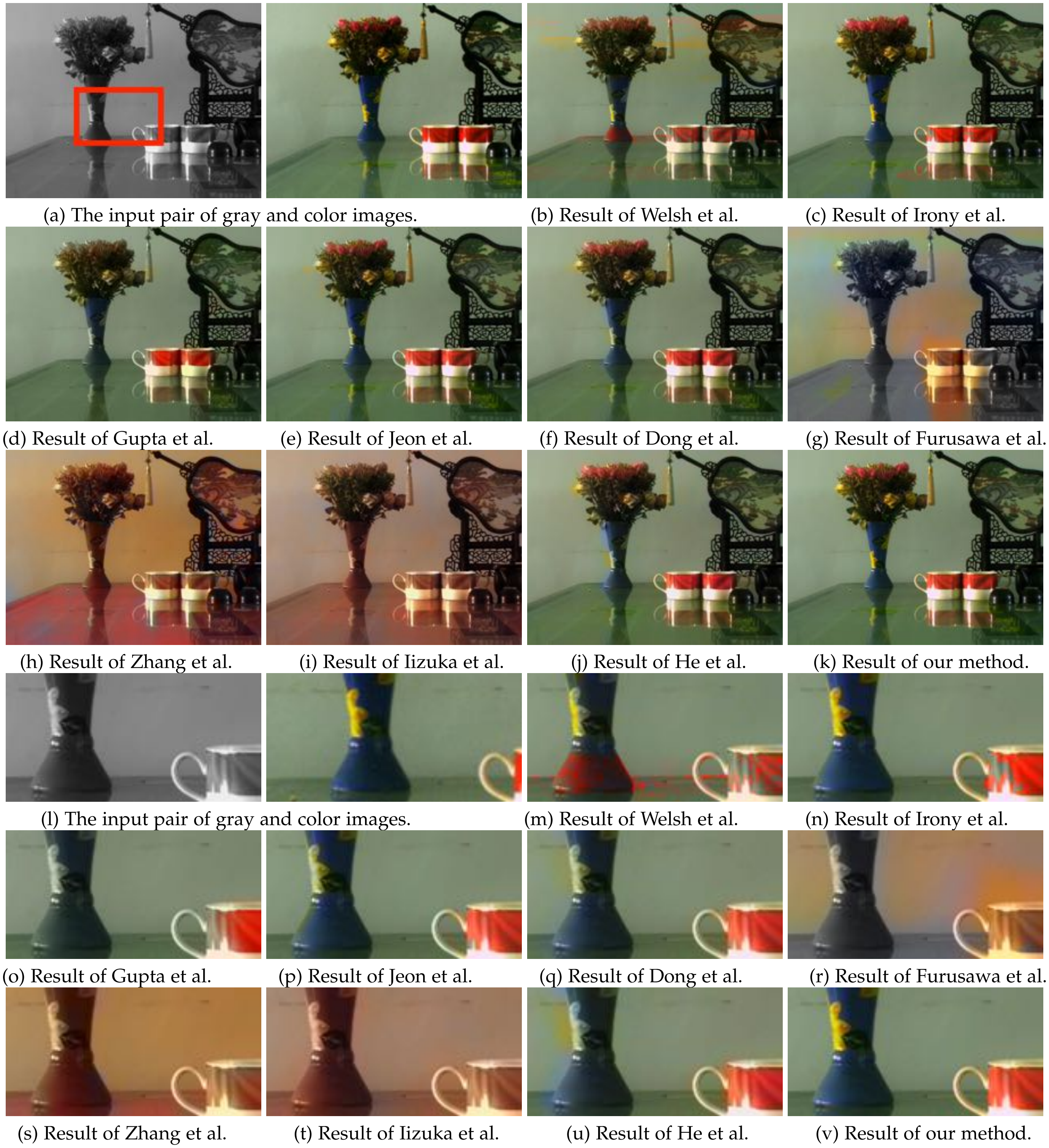
(h) Result of Zhang et al. (i) Result of Iizuka et al. (j) Result of He et al. (k) Result of our method.

(l) The input pair of gray and color images. (m) Result of Welsh et al. (n) Result of Irony et al.

(o) Result of Gupta et al. (p) Result of Jeon et al. (q) Result of Dong et al. (r) Result of Furusawa et al.

(s) Result of Zhang et al. (t) Result of Iizuka et al. (u) Result of He et al. (v) Result of our method.

Fig. 12. Examples to compare the colorization results of all the comparison methods and our colorization method. The region marked with the red box is enlarged in the bottom three rows. The input images are captured by the real monochrome-color dual-lens system of Huawei P9 phone.



(a) The input pair of gray and color images. (b) Result of Welsh et al. (c) Result of Irony et al.

(d) Result of Gupta et al. (e) Result of Jeon et al. (f) Result of Dong et al. (g) Result of Furusawa et al.

(h) Result of Zhang et al. (i) Result of Iizuka et al. (j) Result of He et al. (k) Result of our method.

(l) The input pair of gray and color images. (m) Result of Welsh et al. (n) Result of Irony et al.

(o) Result of Gupta et al. (p) Result of Jeon et al. (q) Result of Dong et al. (r) Result of Furusawa et al.

(s) Result of Zhang et al. (t) Result of Iizuka et al. (u) Result of He et al. (v) Result of our method.

Fig. 13. Examples to compare the colorization results of all the comparison methods and our colorization method. The region marked with the red box is enlarged in the bottom three rows. The input images are captured by the real monochrome-color dual-lens system of Huawei P9 phone.

dual-lens system of Huawei P9 phone in Figs. 12 and 13. As shown, our method can obtain better results than all the comparison methods. Some colorization results of images captured by the real dual-lens system under low-light conditions are also shown in Fig. 11.

5.4 Experiment II: 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: the weighted average of colors of candidate pixels, the attention operation, and the color correction module. The datasets used in this experiment are under Setup1 in Table 2. All the models are trained on the Scene Flow dataset, and tested on the Cityscapes, Middlebury, and Sintel datasets. Table 6 shows the summary performance of different models.

First, we study the differences between our problem and stereo matching. As mentioned in Sec. 2, it is possible to first estimate the disparity between the input image and reference image, and then warp the colors of the reference image according to the estimated disparity to get the colorization result. We implement the state-of-the-art stereo matching method [20], and the results are shown in 'Stereo matching model' of Table 6. Specifically, compared with our model, this model does not have the operation of weighted average, color correction and the attention operation. In addition, it is trained using the ground truth disparity values. As shown in Table 6, its performance is much lower than our model. The reason is that it aims at estimating disparities, but, in the reference image, pixels with wrong disparity values may have correct colors, especially in textureless and repeated texture regions. Moreover, pixels with correct disparity values may have wrong colors, especially in occlusion regions. In short, the problems of colorization and stereo matching are different and therefore need different methods to solve them.

Second, we evaluate the contribution of the weighted average operation. In 'No weighted average', instead of weighted average, we perform soft argmax after getting the weight volume to obtain the best-matching candidate pixel for each pixel, and copy its color as the rough colorization result. As shown in Table 6, its performance is lower than our model, because the weighted average operation could make use of colors of more pixels in the reference image.

Third, we evaluate the contribution of the attention operation. In 'No attention', we do not perform the attention operation and directly use the concatenated feature volume as the input of the 3-D regulation. The results are not as good as our model either.

Fourth, we evaluate the contribution of the color correction module of our model. In 'No color correction', we output the rough colorization result directly as the final result, without performing the color correction module. As shown, the performance decreases a lot without the color correction. It is because the input gray image can provide guidance of spatial color consistency. Using the guidance, the color correction module could correct wrongly colorized pixels by their neighboring pixels.

Fifth, we evaluate the contribution of the improved parts of our method comparing with the conference version of our paper, i.e. Dong et al. [10]. The results of the conference version are shown in the 'Dong et al. [10]' term. As shown, our method achieves higher colorization accuracy due to two main revisions. 1) We improve the color correction part in the colorization CNN and 2) the colorization CNN is used twice in the colorization module and the colorization quality estimation module respectively. To better analyze the contribution of each of the two revisions, we also use the improved color correction part to perform the experiment without the colorization quality estimation module, and the results are shown in the 'Only colorization module' term. From these results, we notice that both revisions have positive effects on the performance. The reasons are that 1) in the color correction part, our goal is to learn the color residue between $\mathbf{C}'$ and the ground-truth colors, as shown in Fig. 3. Transferring $\mathbf{C}'$ to a ResNet feature, which is done in [10], is not a must-do step. Adding the unnecessary ResNet subset into the whole framework will increase the difficulty for training and thus have a negative effect on the prediction accuracy. 2) Comparing with Dong et al. [10], in the training step of our method, we use both the colorization module and the colorization quality estimation module to help train the colorization CNN. So the colorization CNN is trained with more data, which helps improve the performance.

Sixth, we perform an ablation study to use the simple average operation instead of the weighted average operation for the colorization. In detail, for each pixel in the input image, we use the simple average operation to average the colors of all the candidate pixels within the search range of the reference image and use the simple average result as the colorization result. The results are shown in the 'simple average' term of Table 6. As shown, the accuracy is very low. It is because, the search range in the reference image consists of more than one hundred pixels (the search range d is set as 20% of image width in this paper). Although among the set of candidate pixels, there may exist multiple pixels with correct colors, most of them have wrong colors and thus the simple average operation will generate wrong colors in most cases, leading to low PSNR and SSIM values.

Last, we use the ground-truth disparity for colorization, i.e. we use the ground-truth disparity to warp the reference color image and directly copy the color of pixels of the warped color image for the corresponding pixels of the input gray image. As shown in the 'ground-truth disparity' term of Table 6, the accuracy is even lower than many colorization methods, e.g. Jeon et al., He et al., Dong et al., and ours in Tables 4, 5. The reason is that although in non-occlusion regions, the ground-truth disparity can always help obtain correct colors, in occlusion regions, for pixels of the input image, their corresponding pixels in the reference image are missing due to occlusions and the computed corresponding pixels in the reference image using the ground-truth disparity belong to different objects and are usually with different colors. The errors in occlusion regions lead to low PSNR and SSIM values.

5.5 Experiment III: Inlier and outlier judgement

As described in Sec. 4, we estimate the colorization qualities of the colorization results of the colorization module and divide them into inlier and outlier colorization results. The

TABLE 7
Processing time (ms) of different methods for images with different resolutions. The non-learning based methods including Welsh [15], Ironi [2], Gupta [3] and Jeon [1] are run on CPU, and the deep learning based methods including Furusawa [16], He [18], Zhang [11], Iizuka [12], Dong [19] and Ours are run on GPU.

	Welsh	Ironi	Gupta	Jeon	Furusawa	He	Zhang	Iizuka	Dong	Ours
2048 × 1600	2096	18038	183582	354263	31017	13514	6853	4826	577	3062
1024 × 800	1179	4249	36184	93662	8149	3184	1816	1425	153	726
512 × 400	530	969	9249	23861	2361	713	503	358	58	229
256 × 200	267	283	2495	6018	636	154	124	83	19	77

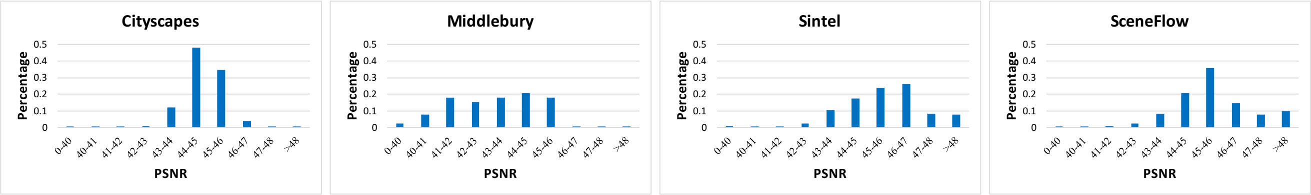


Fig. 14. The distributions of PSNR values of inlier colorization results over the four datasets, i.e. Cityscapes [7], Middlebury [8], Sintel [9], and SceneFlow [6].

TABLE 8
The percentage of inlier and outlier colorization results of the four datasets, i.e. Cityscapes, Middlebury, Sintel, and Scene Flow, using the colorization CNN. The inliers and outliers are judged according to our colorization quality estimation module.

	Cityscapes	Middlebury	Sintel	Scene Flow
Percentage of inliers	99.98%	86.36%	88.82%	99.06%
Percentage of outliers	0.02%	13.64%	11.18%	0.94%

TABLE 9
Average PSNR (dB) values of inlier and outlier colorization results of the four datasets, i.e. Cityscapes, Middlebury, Sintel, and Scene Flow.

	Cityscapes	Middlebury	Sintel	Scene Flow
Average PSNR of inliers	44.79	43.25	45.34	45.66
Average PSNR of outliers	40.71	37.13	36.71	43.30

percentage of inliers and outliers over the four datasets, i.e. Cityscapes, Middlebury, Sintel, and Scene Flow, is shown in Table 8. As shown, most colorization results are inliers, which reflects the effectiveness of the proposed colorization module. In addition, the percentage of inliers and outliers varies among different datasets. It is because the images in different datasets have different levels of occlusions. More occlusions usually lead to lower accuracy.

Besides Table 8, we calculate the average PSNR values of inlier and outlier colorization results of the four datasets, which is shown in Table 9. As shown, the inliers have very high PSNR values on average while the outliers have much lower PSNR values. This reflects that the proposed colorization quality estimation module is effective to judge whether the colorization result is an inlier or an outlier.

From Table 8 and 9, we can find that the combination of the colorization module and the colorization quality estimation module can successfully colorize most input pairs and accurately judge the colorization results into inliers and outliers.

We also show the distributions of PSNR values of inlier colorization results over the four datasets in Fig. 14. As shown, among the inlier results, most images have very high PSNR values and only very few results' PSNR values are less than 40 dB. This figure further reflects that the inlier colorization results according to our framework always have very high quality.

Some example of the outlier colorization results are shown in Fig. 15. The error colors are introduced due to complicated occlusions. And more examples of the inlier colorization results are shown in Fig. 16.

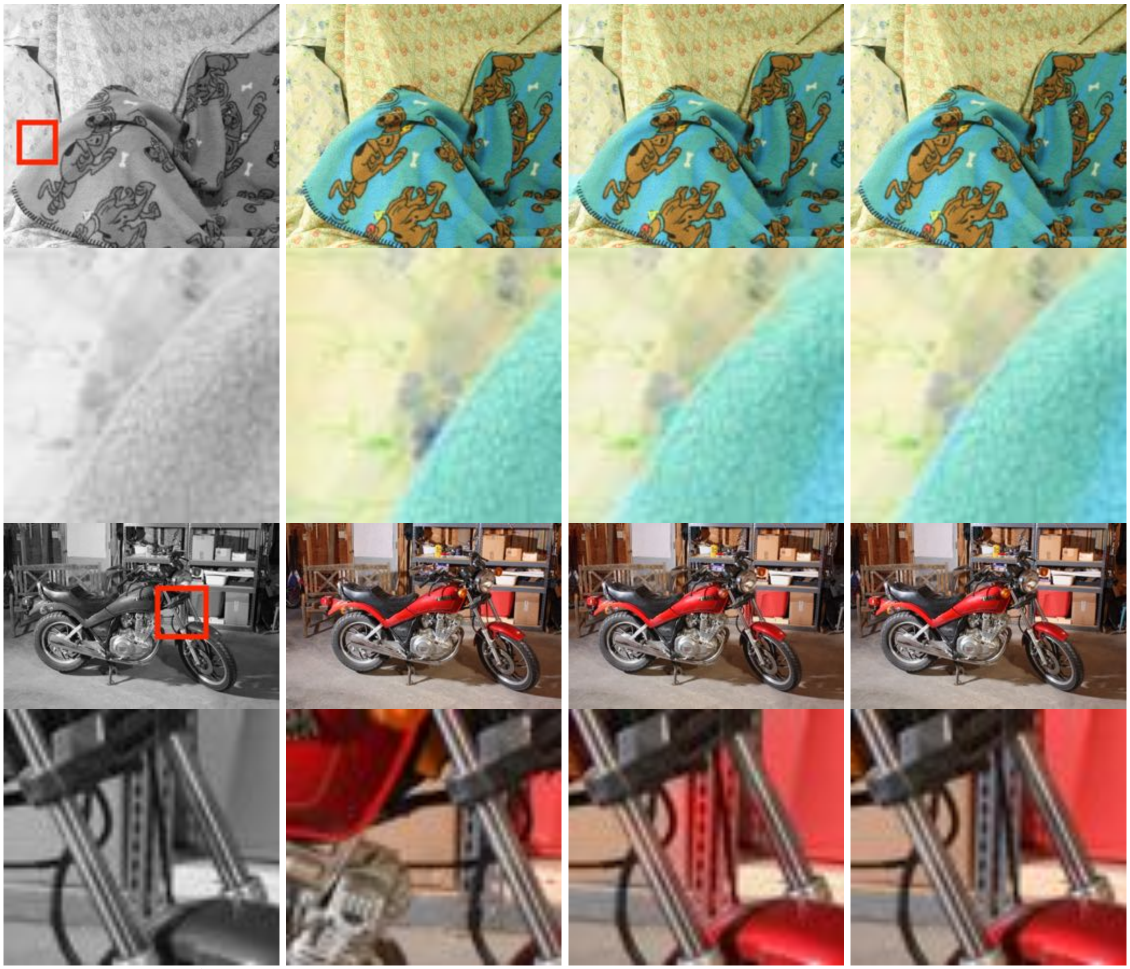
6 Conclusion and Discussion

We have presented a novel deep learning based framework for colorization in monochrome-color dual-lens system, which consists of the colorization module and the colorization quality estimation module. 1) In the colorization module, we propose the colorization CNN which performs weighted average of colors of candidate pixels in the reference image to obtain the colorization result for each pixel in the input gray image. When learning the weight values, we perform the attention operation and 3-D regulation. To correct the results in occlusion regions, we propose the color correction part. 2) In the colorization quality estimation module, based on the symmetry property of colorization, we use the colorization CNN again with horizontal flips for the inputs and outputs to colorize the gray map of the original reference color image using the result of the colorization module as reference. Then, we evaluate the second-time colorization result with the original reference color image as ground-truth, and use the evaluated quality to estimate the colorization quality of the colorization CNN. Experimental results show that our method achieves superior colorization qualities than the state-of-the-art colorization methods, and we can also accurately estimate the colorization quality and divide the colorization results into inliers and outliers.

The computational complexity of the proposed algorithm is not low enough to be easily employed in the smart phones for real-time processing. We will study how to accelerate the algorithm in the future.

Acknowledgments

This work is funded by the National Nature Science Foundation of China (No. 61802026 and 61806016) and the Fun-



(a) The input pair of grayscale and color images. (b) Our result. (c) Ground truth.

Fig. 15. Examples of the outlier results of the colorization module. The marked region with red box is shown in the following rows. As shown, the colorization CNN fails to recover the correct colors of the marked region due to complicated occlusions.

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

[3] 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.

[4] Z. Wang, A. C. Bovik, H. Sheikh, and E. P. Simoncelli, "Image quality assessment from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2016.

[6] 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.

[7] 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.

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

[9] 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.

[10] X. Dong, W. Li, X. Wang, and Y. Wang, "Learning a deep convolutional network for colorization in monochrome-color dual-lens system," *AAAI*, pp. 1–9, 2019.

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

[12] 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, pp. 1–15, 2016.

[13] 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, pp. 1–11, 2017.

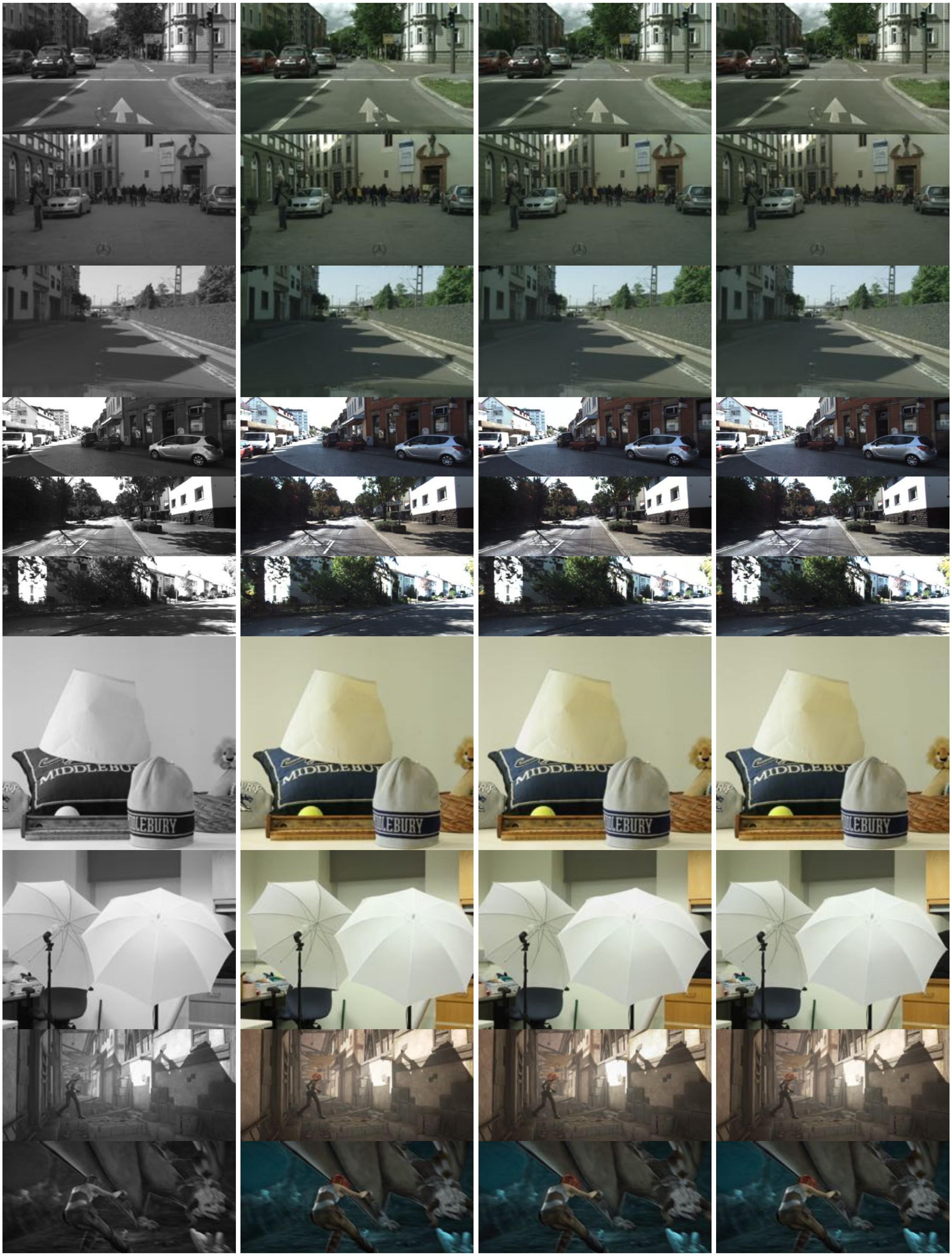
[14] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 689–694, 2004.

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

[16] C. Furusawa, K. Hiroshiba, K. Ogaki, and Y. Odagiri, "Comicolorization: semi-automatic manga colorization," *SIGGRAPH Asia*, pp. 1–11, 2017.

[17] M. He, J. Liao, L. Yuan, and P. Sander, "Neural color transfer between images," *Arxiv*, pp. 1–18, 2017.

[18] M. He, D. Chen, J. Liao, P. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM SIGGRAPH*, pp. 1–11, 2018.



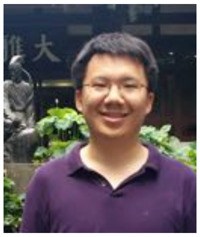
(a) The input pair of grayscale and color images.

(b) Our result.

(c) Ground truth.

Fig. 16. Examples of the inlier results of the colorization module.

[19] X. Dong and W. Li, "Shoot high-quality color images using dual-lens system with monochrome and color cameras," *Neurocomputing*, no. 352, pp. 22–32, 2019.
[20] 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*, pp. 1–9, 2017.
[21] B. Li, C. Lin, B. Shi, T. Huang, W. Gao, and C. Kuo, "Depth-aware stereo video retargeting," *CVPR*, 2018.
[22] D. Jeon, S. Baek, I. Choi, and M. Kim, "Enhancing the spatial resolution of stereo images using a parallax prior," *CVPR*, 2018.
[23] 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.
[24] S. Zhou, J. Zhang, W. Zuo, H. Xie, J. Pan, and J. Ren, "Davanet: Stereo deblurring with view aggregation," *CVPR*, 2019.
[25] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," *CVPR*, 2017.
[26] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stereoscopic neural style transfer," *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2018.
[27] L. Pan, Y. Dai, M. Liu, and F. Porikli, "Simultaneous stereo video deblurring and scene flow estimation," *CVPR*, 2017.
[28] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2018.
[29] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.
[30] P. Ye, J. Kumar, and D. Doermann, "Beyond human opinion scores: Blind image quality assessment based on synthetic scores," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4241–4248, 2014.
[31] Z. Lin, M. Feng, C. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *International Conference on Learning Representations*, pp. 1–15, 2017.
[32] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," *International Conference on Neural Information Processing Systems*, pp. 1–11, 2016.
[33] Y. Li, J. Huang, N. Ahuja, and M. Yang, "Deep joint image filtering," *European Conference on Computer Vision*, pp. 1–14, 2016.
[34] J. Kim, J. Lee, and K. Lee, "Accurate image super-resolution using very deep convolutional networks," *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2016.
[35] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Multiplexing for optimal lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1339–1354, 2007.
[36] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, 2012.
[37] S. Duggal, S. Wang, W.-C. Ma, R. Hu, and R. Urtasun, "Deeppruner: Learning efficient stereo matching via differentiable patchmatch," 2019.
[38] J. Chen, A. Adams, N. Wadhwa, and S. W. Hasinoff, "Bilateral guided upsampling," *ACM SIGGRAPH ASIA*, 2016.



Xuan Dong received the Ph.D. degree in Computer Science from Tsinghua University in 2015, and the B.E. degree in Computer Science from Beihang University in 2010. He is currently an Associate Professor in the School of Computer Science, Beijing University of Posts and Telecommunications, China. His research interests include computer vision, computer graphics and computational photography.

Weixin Li received the Ph.D. degree in computer science from the University of California at Los Angeles (UCLA), Los Angeles, CA, USA, in 2017. She is currently an Associate Researcher at the School of Computer Science and Engineering (SCSE) and Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), Beihang University, Beijing, China. Her research interests include computer vision, image processing, and big data analytics.

Xiaoyan Hu is currently pursuing the M.E. degree in Computer Technology at Beijing University of Posts and Telecommunications. She received the B.E. degree in Computer Science and Technology from Beijing University of Posts and Telecommunications in 2019. Her current research interests include computer vision, computer graphics and computational photograph, with a focus on the study of high dynamic range imaging.

Xiaojie Wang received his Ph.D. degree from Beihang University in 1996. He is a professor and director of the Centre for Intelligence Science and Technology at Beijing University of Posts and Telecommunications. His research interests include natural language processing and multi-modal cognitive computing. He is an executive member of the Council of Chinese Association of Artificial Intelligence, director of Natural Language Processing Committee. He is a member of Council of Chinese Information Processing Society and Chinese Processing Committee of China Computer Federation.

Yunhong Wang received the B.S. degree in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 1989, and the M.S. and Ph.D. degrees in electronic engineering from Nanjing University of Science and Technology, Nanjing, China, in 1995 and 1998, respectively. She was with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, from 1998 to 2004. Since 2004, she has been a Professor with the School of Computer Science and Engineering, Beihang University, Beijing, where she is also the Director of Laboratory of Intelligent Recognition and Image Processing, Beijing Key Laboratory of Digital Media. Her current research interests include biometrics, pattern recognition, computer vision, data fusion, and image processing.