The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Learning a Deep Convolutional Network for Colorization in Monochrome-Color Dual-Lens System

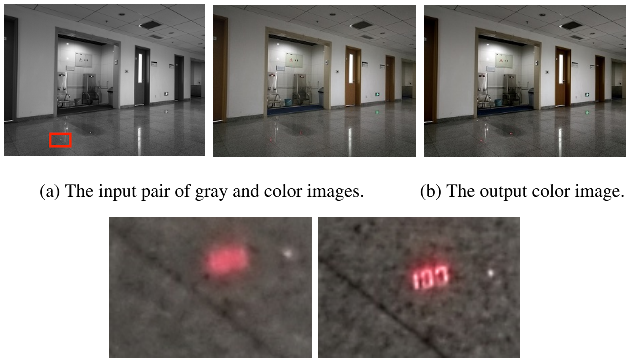
Xuan Dong,[1] Weixin Li,[2]* Xiaojie Wang,[1] Yunhong Wang[2]
[1]School of Computer Science, Beijing University of Posts and Telecommunications,
[2]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University
dongxuan8811@bupt.edu.cn, weixinli@buaa.edu.cn, xjwang@bupt.edu.cn, yhwang@buaa.edu.cn

Abstract

In the monochrome-color dual-lens system, the gray image captured by the monochrome camera has better quality than the color image from the color camera, but does not have color information. To get high-quality color images, it is desired to colorize the gray image with the color image as reference. Related works usually use hand-crafted methods to search for the best-matching pixel in the reference image for each pixel in the input gray image, and copy the color of the best-matching pixel as the result. We propose a novel deep convolution network to solve the colorization problem in an end-to-end way. Based on our observation that, for each pixel in the input image, there usually exist multiple pixels in the reference image that have the correct colors, our method performs weighted average of colors of the candidate pixels in the reference image to utilize more candidate pixels with correct colors. The weight values between pixels in the input image and the reference image are obtained by learning a weight volume using deep feature representations, where an attention operation is proposed to focus on more useful candidate pixels and a 3-D regulation is performed to learn with context information. In addition, to correct wrongly colorized pixels in occlusion regions, we propose a color residue joint learning module to correct the colorization result with the input gray image as guidance. We evaluate our method on the Scene Flow, Cityscapes, Middlebury, and Sintel datasets. Experimental results show that our method largely outperforms the state-of-the-art methods.

Introduction

The dual-lens system with one monochrome camera and one color camera has been widely used in popular smart phones, e.g. Huawei P9 and P10. In the dual-lens system, the monochrome camera has better light efficiency than the color camera (Jeon et al. 2016), so the image captured by the monochrome camera has higher quality (i.e. signal-to-noise ratio) than the image from the color camera, but does not have color information. To shoot high quality color images using dual-lens systems, it is desirable to colorize the gray images from the monochrome camera with the color images from the color camera as reference, so that the colorized images have high quality in the monochrome channel and correct colors as well. An example is shown in Fig. 1.



(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.

Figure 1: An example of colorization in the dual-lens system. The input images are captured by the dual-lens system of Huawei P9 phone. The output colorization result has high quality in the monochrome channel and correct colors.

In the literature, reference-based colorization methods, e.g. (Ironi, Cohen-Or, and Lischinski 2005), (Gupta et al. 2012), (Jeon et al. 2016), are related to our problem. Most methods, e.g. (Ironi, Cohen-Or, and Lischinski 2005), (Gupta et al. 2012), 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. Jeon et al. (Jeon et al. 2016) use a stereo matching method, which is based on brightness constancy and edge similarity constraints, to search for the best-matching pixel. For occlusion pixels, color propagation, e.g. (Levin, Lischinski, and Weiss 2004), is usually performed to estimate or correct the colors of these pixels based on their neighboring pixels. Recently, deep learning based methods have proven to be effective for many vision problems compared with traditional 'hand-crafted' methods, e.g. single image super-resolution (Dong et al. 2014), and stereo matching (Zbontar and LeCun 2016). However, for reference-based colorization, to the best of our knowledge, deep learning based methods have not been explored yet. In addition, to estimate the color of each pixel, previous methods usually copy the color of only one pixel in the reference image as the result. We

*Corresponding Author.

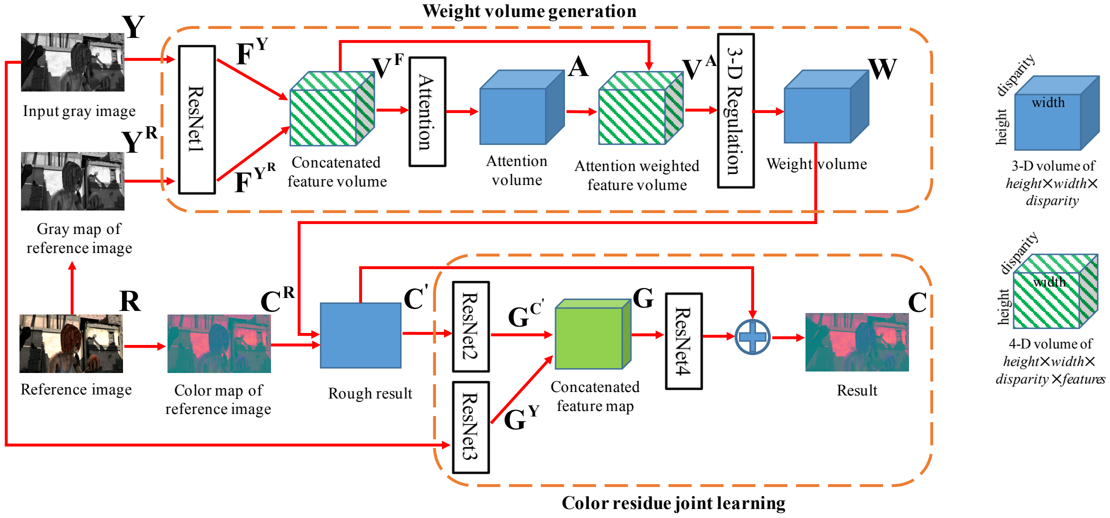


Figure 2: The overall structure of our model. (Best viewed in color)

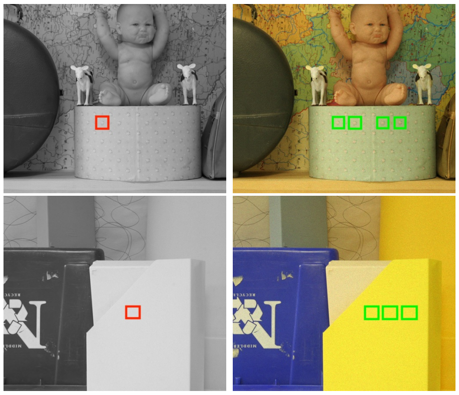


Figure 3: Examples to show there usually exist several similar pixels (marked in green) in the reference image that could provide correct colors for a given pixel (marked in red) in the input gray image.

notice that, as shown in Fig. 3, for each 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 noises and diminish errors in occlusion regions.

To deal with these issues, in this paper, we propose a convolutional neural network that solves the colorization problem in an end-to-end way. 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.

The framework of our method is shown in Fig. 2. 1) 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, first, we extract deep features of the input gray image and the gray map of reference image by ResNet (He et al. 2015), and build a concatenated feature volume. Then, to get 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 and thus get the attention weighted feature volume. Next, to estimate the weight values with context information, we use the 3-D regulation to compute the weight volume. 2) After getting the weight volume, we perform weighted average using the estimated weight and the reference color map to get the rough colorization result. 3) The 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 residue joint learning module to correct the colorization result with the input gray image as guidance.

Experimental results show that the proposed method largely outperforms the state-of-the-art algorithms in four datasets, including Scene Flow (Mayer et al. 2016), Cityscapes (Cordts et al. 2016), Middlebury (Scharstein and Pal 2007), and Sintel (Butler et al. 2012).

Our contributions include: 1) For estimating the color of 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 convolutional network, attention mechanism, 3-D regulation and color residue joint learning are used for improving the accuracy of weight estimation and correcting colors in occlusion regions. 3) We achieve the highest accuracy in all the four datasets compared with the state-of-the-art algorithms.

Related Work

In the literature, there exist three kinds of colorization algorithms, including automatic colorization, scribble-based colorization, and reference-based colorization.

Automatic colorization algorithms, e.g. (Zhang, Isola, and Efros 2016) and (Iizuka, Simo-Serra, and Ishikawa 2016), directly colorize gray images without any reference. Using them in our problem is not proper because the reference color image, which provides much useful color information, will not be utilized.

Scribble-based colorization algorithms, e.g. (Zhang et al. 2017) and (Levin, Lischinski, and Weiss 2004), need users to input some scribbles or strokes as guidance for colorization. Because user input is not available in the camera system, these algorithms are not suitable for our problem.

Reference-based colorization algorithms, e.g. (Welsh, Ashikhmin, and Mueller 2002; Ironi, Cohen-Or, and Lischinski 2005; Gupta et al. 2012; Jeon et al. 2016; Furusawa et al. 2017; He et al. 2017; 2018), are related to our problem. Welsh et al. (Welsh, Ashikhmin, and Mueller 2002) 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. (Ironi, Cohen-Or, and Lischinski 2005) 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 (Levin, Lischinski, and Weiss 2004). Gupta et al. (Gupta et al. 2012) 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. Jeon et al. (Jeon et al. 2016) search for best-matching pixels by a stereo matching method, which is based on brightness constancy and edge similarity constraints, and correct colors in occlusion regions by applying spatial consistency of neighboring pixels over the whole image. Furusawa et al. (Furusawa et al. 2017) 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. (He et al. 2018) propose a deep learning based algorithm. But, they 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 loss minimizes the semantic differences of unnatural colorization. The result looks natural but is not always faithful to the ground truth colors.

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. (Alex et al. 2017), to estimate the disparity between the images, and then copy colors of the corresponding pixels in the reference image to 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 stereo system, like style transfer(Chen et al. 2018). But, these methods cannot be directly used for our problem.

Method

The framework of our model is shown in Fig. 2. 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. Second, we use it to perform the weighted average operation and obtain the rough colorization result. Third, we perform the color residue joint learning to correct the wrongly colorized pixels in occlusion regions.

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 big, noises or even errors will be introduced to the colorization results. We propose an attention operation to reduce the noises/errors. Attention mechanism has been successfully used in various problems, e.g. text classification (Lin et al. 2017), and visual question answering (Lu et al. 2016). 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 obtain higher weight values of useful candidate pixels and reduce noises/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 (Alex et al. 2017), 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 residue joint learning module in our network. We share similar insights with (Levin, Lischinski, and Weiss 2004) that neighboring pixels with similar gray intensities should have similar colors, and the input gray image $\mathbf{Y}$ could provide guidance of spatial color consistency. Our method is based on the deep joint filter (Li et al. 2016). Our difference from (Li et al. 2016) is that 1) we use ResNet (He et al. 2015) instead of traditional 2-D convolution due to good performances of ResNet in related problems, and 2) 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 (Kim, Lee, and Lee 2016).

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 $\mathbf{C} \in \mathbb{R}^{h \times w}$ of the input gray image $\mathbf{Y} \in \mathbb{R}^{h \times w}$ from the monochrome camera. We use the *YCbCr* color space in this paper. The *Y* channel map of $\mathbf{R}$ is denoted as $\mathbf{Y}^{\mathbf{R}}$. The *Cb* and *Cr* channel maps are predicted respectively. So $\mathbf{C}^{\mathbf{R}}$ denotes the *Cb*/*Cr* channel map of the reference image, and $\mathbf{C}$ denotes the corresponding predicted *Cb*/*Cr* channel map. 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.

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

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 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 $\mathbf{Y}$ as guidance to correct the rough result $\mathbf{C}^{'}$ by

$$\mathbf{C} = \mathbf{C}' + \Phi(\mathbf{C}', \mathbf{Y}), \quad (2)$$

where Φ denotes the operation of the color residue joint learning.

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. 2. The inputs include the input gray image $\mathbf{Y}$ and the gray map of the reference image $\mathbf{Y}^{\mathbf{R}}$.

First, we extract the deep features $\mathbf{F}^{\mathbf{Y}} \in \mathbb{R}^{h \times w \times n}$ and $\mathbf{F}^{\mathbf{Y}^{\mathbf{R}}} \in \mathbb{R}^{h \times w \times n}$ of $\mathbf{Y}$ and $\mathbf{Y}^{\mathbf{R}}$ 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 $\mathbf{F}^{\mathbf{Y}}_{j,i}$ with features of each candidate pixel $\mathbf{F}^{\mathbf{Y}^{\mathbf{R}}}_{j,i+k}$. And the concatenated features of all pixels and their candidate pixels form the 4-D feature volume $\mathbf{V}^{\mathbf{F}} \in \mathbb{R}^{h \times w \times d \times 2n}$, where

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

Next, the attention operation, which consists of two 3-D convolution layers, is performed to obtain the attention volume $\mathbf{A}$ from the feature volume $\mathbf{V}^{\mathbf{F}}$. 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 $\mathbf{V}^{\mathbf{F}}$ by

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

Next, the 3-D regulation, which is proposed by (Alex et al. 2017) to learn with context, is performed to estimate the weight volume $\mathbf{W}$ from the attention weighted feature volume $\mathbf{V}^{\mathbf{A}}$.

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

Table 1: Summary of our deep colorization architecture. 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 $\mathbf{Y}^{\mathbf{R}}$	$h \times w$
	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	
58-75	repeat layers 40-57	$h \times w \times n$
	ResNet4	
76-92	repeat layers 40-56	$h \times w \times n$
93	3×3 conv, 1 feat. (no ReLu, BN)	$h \times w$

Color residue joint learning

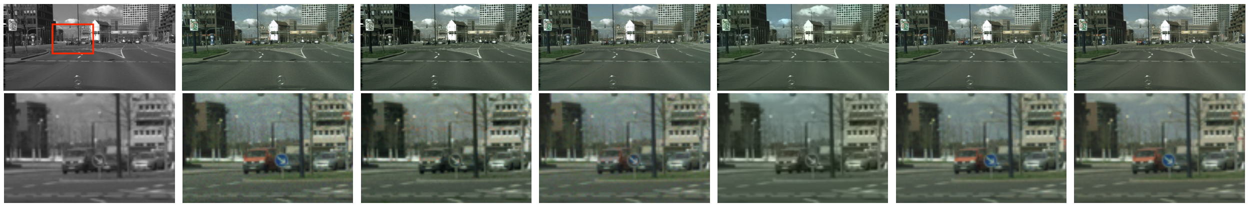
Our goal is to use the input gray image $\mathbf{Y}$ as guidance to correct the rough colorization result $\mathbf{C}'$, which may contain wrongly colorized pixels due to occlusions.

As shown in Fig. 2, the rough colorization result $\mathbf{C}'$ and the input gray image $\mathbf{Y}$ are fed into two ResNets, named ResNet2 and ResNet3, to get their features $\mathbf{G}^{\mathbf{C}'}$ and $\mathbf{G}^{\mathbf{Y}}$ respectively. ResNet2 and ResNet3 have the same structure but the parameters are trained separately. Then, $\mathbf{G}^{\mathbf{C}'}$ and $\mathbf{G}^{\mathbf{Y}}$ are concatenated to form the feature map $\mathbf{G}$, which is fed into another ResNet, named ResNet4, to get the residue color map $\Phi(\mathbf{C}', \mathbf{Y})$. By adding $\mathbf{C}'$ and the residue color map $\Phi(\mathbf{C}', \mathbf{Y})$, the final colorization result $\mathbf{C}$ is obtained. The color residue joint learning module can be seen as a high dimension joint filter.

Network architecture

We show our network architecture in Fig. 2. 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 ker-



(a) The input pair of gray and color images. (b) Result of Welsh et al. (c) Result of Ironi et al. (d) Result of Gupta et al. (e) Our result. (f) Ground truth.

Figure 4: An example to compare the colorization results of Welsh et al.'s method, Ironi et al.'s method, Gupta et al.'s method, and our colorization method. The region marked with the red box is shown in the second row. As shown, the comparison methods fail to recover correct colors in the marked region. This example is under Setup1 in Table 2.

Table 2: Two setups of the colorization benchmark. We simulate the monochrome-color dual-lens system by adding signal dependent Gaussian noises with a given standard deviation where κ represents the noise-free signal intensity (Achanta et al. 2007).

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

nel and stride 2. Here, we downsample the data with stride 2 to reduce memory cost. The resolution is recovered 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 (Alex et al. 2017) for more details. 2) In the color residue joint learning module, we use three ResNets, named ResNet2, ResNet3 and ResNet4. 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. ResNet3 has the same network structure as ResNet2. The difference between ResNet4 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, ResNet3, and ResNet4 are trained separately.

Experiments

Datasets

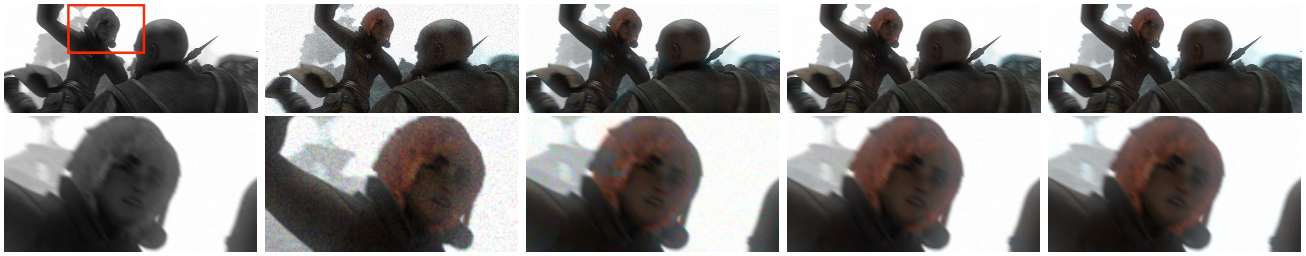
We use four popular stereo datasets in our experiments, namely Cityscapes (Cordts et al. 2016), Middlebury (Scharstein and Pal 2007), Sintel (Butler et al. 2012), and SceneFlow (Mayer et al. 2016). These datasets contain pairs of color images captured by the dual-lens system with two color cameras. For realistic simulations, following (Jeon et al. 2016), within each pair of images, we de-color one image and use the de-colored result as the input monochrome image, and the other color image is used as the input color image. In addition, we imitate the light-efficiency differences between color and monochrome cameras by adding different amount of noises to the monochrome input images and color input images. We configure two different setups for this experiment. The details are summarized in Table 2.

Implementation details

The proposed deep convolutional network is implemented with TensorFlow. All models are optimized with RMSProp (Tieleman and Hinton 2012) 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. The loss function we use is the mean squared error between the prediction results and the ground-truth color maps. When testing the performance on the other three datasets, we directly use the model trained on Scene Flow for cross-validation.

Experiment I: Comparison with other colorization methods

Comparison algorithms: First, we compare with state-of-the-art reference-based colorization algorithms, i.e. the methods of Welsh et al. (Welsh, Ashikhmin, and Mueller 2002), Ironi et al. (Ironi, Cohen-Or, and Lischinski 2005), Gupta et al. (Gupta et al. 2012), Jeon et al. (Jeon et al. 2016), Furusawa et al. (Furusawa et al. 2017) and He et al. (He et al. 2018). In addition, we compare with two state-of-the-art deep learning based automatic colorization algorithms, i.e. the methods of Zhang et al. (Zhang, Isola, and Efros 2016) and Iizuka et al. (Iizuka, Simo-Serra, and Ishikawa



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

(b) Result of Jeon et al.

(c) Our result.

(d) Ground truth.

Figure 5: An example to compare the colorization results of Jeon et al.'s method 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.

Table 3: 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	35.06	29.60	32.70	29.83
Ironi	38.45	32.98	36.06	31.24	35.68	30.42	32.52	30.56
Gupta	38.09	31.04	35.45	29.65	34.53	30.72	33.31	29.73
Jeon	39.33	36.80	36.12	31.32	35.38	34.75	33.98	31.78
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
Ours	**44.26**	**41.94**	**43.88**	**45.18**	**43.21**	**40.30**	**42.71**	**44.16**

Table 4: Average SSIM values of different colorization methods in four datasets under Setup 1 and 2 in Table 2.

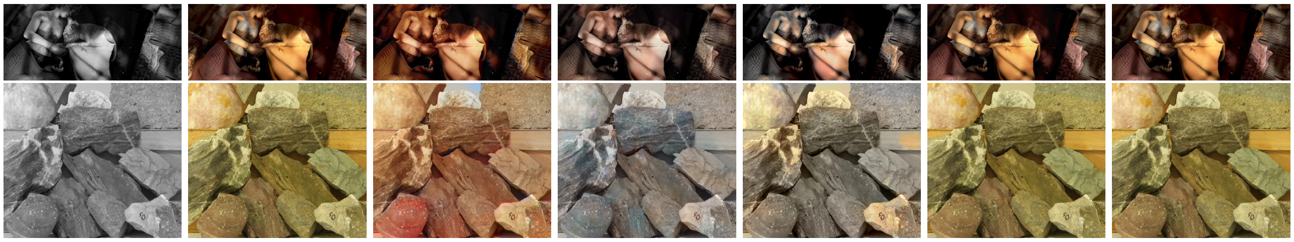
	SSIM under Setup1				SSIM under Setup2			
	CT	MB	ST	SF	CT	MB	ST	SF
Welsh	0.897	0.906	0.795	0.813	0.849	0.876	0.758	0.769
Ironi	0.897	0.940	0.918	0.890	0.778	0.715	0.814	0.747
Gupta	0.948	0.896	0.933	0.869	0.906	0.893	0.905	0.750
Jeon	0.953	0.958	0.943	0.927	0.914	0.953	0.924	0.902
Furusawa	0.841	0.860	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
Ours	**0.982**	**0.981**	**0.983**	**0.988**	**0.979**	**0.976**	**0.977**	**0.984**

Table 5: 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 residue joint learning	36.04	37.55	36.05	35.98	0.954	0.955	0.957	0.959
Ours	**44.26**	**41.94**	**43.88**	**45.18**	**0.982**	**0.981**	**0.983**	**0.988**

2016), which could automatically colorize monochrome images without any reference images. The methods of Welsh et al. (Welsh, Ashikhmin, and Mueller 2002), Ironi et al. (Ironi, Cohen-Or, and Lischinski 2005), and Gupta et al. (Gupta et al. 2012) 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 the section of Formulation. 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 3 and 4. As shown, our method largely outperforms the comparison methods. And some qualitative colorization results are shown in Figs. 4, 5, and 6. As shown in Fig. 4, 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 for edges and small objects because many unoccluded pixels are wrongly marked as occluded pixels, 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. 5, 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. 6, 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 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, the perceptual loss minimizes the semantic differences of unnatural colorization. The result



(a) Input gray and color images. (b) Result of Zhang et al. (c) Result of Iizuka et al. (d) Result of Furusawa et al. (e) Our result. (f) Ground truth.

Figure 6: Examples to compare deep learning based automatic colorization algorithms, i.e. Zhang et al. and Iizuka et al., manga image colorization algorithm, i.e. Furusawa et al., 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 assumptions of manga images are not true for general real-world images. The top and bottom examples are from Setup1 and Setup2 in Table 2, respectively.

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 colorization qualities of the state-of-the-art CNN-based automatic colorization methods (Iizuka, Simo-Serra, and Ishikawa 2016; Zhang, Isola, and Efros 2016) are worse than most of the reference-based mathods and ours. As shown in Fig. 6, 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.

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 residue joint learning 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 5 shows the summary performance of different models.

First, we study the differences between our problem and stereo matching. As mentioned in the section of Related Work, 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 (Alex et al. 2017), and the results are shown in 'Stereo matching model' of Table 5. Specifically, compared with our model, this model does not have the operation of weighted average, color residue joint learning and the attention operation. In addition, it is trained using the ground truth disparity values. As shown in Table 5, 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, and 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 performing 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 5, its performance is lower than our model, because the average weighted 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 too.

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

Conclusion

We have presented a novel deep learning method for colorization in monochrome-color dual-lens system. It 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 residue joint learning module. Our method achieves superior performance than the state-of-the-art methods.

Acknowledgments

This work is funded by the National Nature Science Foundation of China (No. 61802026 and 61806016). We thank the anonymous reviewers for helping us to improve this paper.

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Susstrunk, S. 2007. Multiplexing for optimal lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(8):1339–1354.

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

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

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G. 2018. Stereoscopic neural style transfer. *The IEEE Conference on Computer Vision and Pattern Recognition*.

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

Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2014. Learning a deep convolutional network for image super-resolution. *European Conference on Computer Vision* 184–199.

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

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*.

He, M.; Liao, J.; Yuan, L.; and Sander, P. 2017. Neural color transfer between images. *Arxiv*.

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

Iizuka, S.; Simo-Serra, E.; and Ishikawa, H. 2016. 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* 35(4).

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

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

Kim, J.; Lee, J.; and Lee, K. 2016. Accurate image super-resolution using very deep convolutional networks. *The IEEE Conference on Computer Vision and Pattern Recognition*.

Levin, A.; Lischinski, D.; and Weiss, Y. 2004. Colorization using optimization. *ACM transactions on graphics* 23(3):689–694.

Li, Y.; Huang, J.; Ahuja, N.; and Yang, M. 2016. Deep joint image filtering. *European Conference on Computer Vision*.

Lin, Z.; Feng, M.; Santos, C.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *International Conference on Learning Representations*.

Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical question-image co-attention for visual question answering. *International Conference on Neural Information Processing Systems*.

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

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

Tieleman, T., and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*.

Welsh, T.; Ashikhmin, M.; and Mueller, K. 2002. Transferring color to greyscale images. *ACM transactions on graphics* 21(3):277–280.

Zbontar, J., and LeCun, Y. 2016. Stereo matching by training a convolutional neural network to compare image patches. *The Journal of Machine Learning Research* 17(1):2287–2318.

Zhang, R.; Zhu, J.; Isola, P.; Geng, X.; Lin, A.; Yu, T.; and Efros, A. 2017. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics* 36(4).

Zhang, R.; Isola, P.; and Efros, A. 2016. Colorful image colorization. *European Conference on Computer Vision* 649–666.