The Visual Computer manuscript No.
(will be inserted by the editor)

Deep Color Transfer using Histogram Analogy

Junyong Lee · Hyeongseok Son · Gunhee Lee · Jonghyeop Lee · Sunghyun Cho · Seungyong Lee

Abstract We propose a novel approach to transferring the color of a reference image to a given source image. Although there can be diverse pairs of source and reference images in terms of content and composition similarity, previous methods are not capable of covering the whole diversity. To resolve this limitation, we propose a deep neural network that leverages *color histogram analogy* for color transfer. A histogram contains essential color information of an image, and our network utilizes the analogy between the source and reference histograms to modulate the color of the source image with abstract color features of the reference image. In our approach, histogram analogy is exploited basically among the whole images, but it can also be applied to semantically corresponding regions in the case that the source and reference images have similar contents with different compositions. Experimental results show that our approach effectively transfers the reference colors to the source images in a variety of settings. We also demonstrate a few applications of our approach, such as palette-based recolorization, color enhancement, and color editing.

Keywords Color Transfer · Histogram Analogy

1 Introduction

Color transfer is the task of converting a source image according to the reference color information that has desired color characteristics. Many previous methods use guidance images for the reference color information, where the reference images with similar contents and compositions to the source images are needed to obtain visually pleasing results.

J. Lee, H. Son, J. Lee, S. Cho, S. Lee
POSTECH

G. Lee
NCsoft

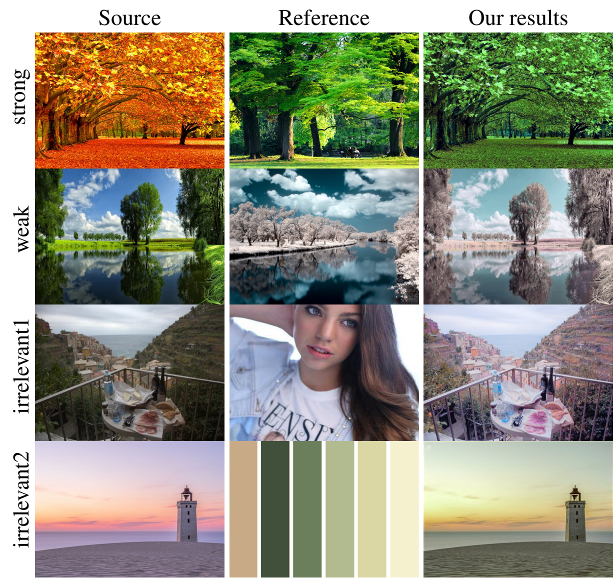

Fig. 1 Color transfer results on various cases.

Such a setting would not be ideal for users as an additional task of searching desired images is required. In this paper, we focus on handling various reference images so that our method can produce plausible color transfer results for diverse pairs of source and reference images (Fig. 1).

We first divide various correlations between source and reference images into three cases (Fig. 1). The first case is the strong relevance, where two images have high similarity both in the contents and positions of semantic objects. Second, the weak relevance refers to high similarity in the contents but with less correlations in the object spatial configurations. The last case of irrelevance includes image pairs with dissimilar contents and special settings with graphic design images and color palettes as the reference.

Traditional global color transfer methods (GCT) [30, 29, 5], which transfer the global color statistics of the refer-

Image correlation	GCT	LCT	Ours
strong relevance	O	O	O
weak relevance	X	O	O
irrelevance	O	X	O

Table 1 Comparison of different approaches. Global color transfer (GCT) may produce less desirable results when the source and target images have similar contents but different object configurations. Local methods (LCT) would not be effective in handling irrelevant source and target images. Our approach uses simple information of color histograms but can handle all cases effectively.

ence image to the source, work well for the case of strong relevance. They can also produce faithful results for irrelevant image pairs, as the overall color distribution is more important in that case. However, in the case of weak relevance, global methods would not effectively reflect semantic correspondences between images and may fail. Local color transfer methods (LCT), which include photo-realistic style transfer [26, 17] and a dense correspondence based approach [8], can successfully handle weakly correlated image pairs. Such methods can also be applied to the case of strong relevance, where the semantic correspondences are rather obvious. Nevertheless, local methods may not produce the best results for the case of irrelevance due to unclear semantic correspondences and less clear style information (Table 1).

For more robust color transfer on various cases, we introduce a novel deep learning based approach using color histograms. Our approach includes two networks; Histogram Encoding Network (HEN) and Color Transfer Network (CTN). Our HEN extracts encoded information from the histograms of source and reference images, which is fed into our CTN to guide the color transfer process. Although a histogram is a simple and global representation of image colors, convolutional neural networks with encoded histograms can conditionally transfer the colors of reference images to source images.

Conceptually, our approach can be interpreted as *histogram analogy* (Fig. 2), in the sense that the correlation between the source and reference histograms is transferred to the source image, similarly to image analogy [9]. Histogram can be extracted from other inputs than ordinary images, such as graphics design images and color palettes. Experimental results demonstrate that histogram analogy provides sufficient information for color transfer in a variety of settings.

In our approach, histogram analogy is exploited either uniformly or adaptively on the source image, depending on the relevance of an input pair. For strongly relevant and irrelevant cases, the same histogram information is used for all parts of the source image, and this is the default setting. When semantic object information is important, as in the case of weak relevance, we can use semantic image segmentation [24] and the histogram analogy is extracted and

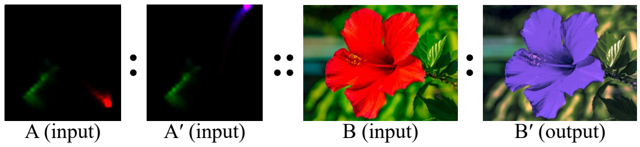

Fig. 2 Histogram analogy. In our histogram analogy, the color correlation between A and A' (the *source* and *reference histograms*, respectively) is transferred to the source image B to obtain the color transfer result B'. The visualization of a histogram shows the 2D normalized histograms of I^{ab} in the Lab color space, where the colors and intensities are determined by the representative colors and occurrences of the histogram bins.

applied for corresponding semantic regions between source and reference images. With the default setting and the variation with semantic segmentation, our approach can support both global and local color transfers, covering the whole diversity of input image pairs.

Our main contributions are:

- We present a robust deep learning based method for color transfer that can produce plausible results for a variety of input images.
- To the best of our knowledge, ours is the first approach that directly utilizes *histogram analogy* in a deep learning network for image color transfer.
- Our framework can deal with various types of inputs for reference colors, such as arbitrary images, color palettes, and color histograms.

2 Related Work

Image analogy For an image B, image analogy [9, 20] generates a converted image B' by transferring the analogy among a given pair of images A and A', achieving $A : A' :: B : B'$. Image analogy has shown dramatic performance on image stylization as well as other applications. In our histogram analogy, A and A' are the source and reference histograms, respectively, and the color correlation between A and A' is transferred to the source image B to obtain the color transfer result B'. Note that the reference image itself is not used in our color transfer framework as we use only its histogram instead. Liao *et al.* [21] consider image analogy in color transfer but their method heavily depends on dense correspondences between the reference and source images at the feature level. In contrast, our approach utilizes abstract information of color histograms to exploit the analogy and can handle more various cases.

Color transfer Early color transfer approaches [30, 29] change colors of a source image based on the global color distribution of a reference image. Reinhard *et al.* [30] perform color transfer by matching the means and standard deviations of source to reference images. Pitie *et al.* [29] explore color mapping between source and reference images leveraging probability density functions of colors. These conventional approaches, however, may fail to reflect semantic correspondence because they do not consider any spatial information.

To handle the limitation of global color transfer, several approaches [31, 25, 10, 1, 32, 8] contemplate local correspondences between source and reference images. Tai *et al.* [31] and Hristova *et al.* [10] employ an EM algorithm to find corresponding clusters in images. Arbelot *et al.* [1] utilize texture descriptors to find similarity maps between source and reference images. He *et al.* [8] obtain more accurate color transfer between objects using neural representations for semantically-meaningful dense correspondences. While these approaches solve the problem of global color transfer in the case that source and reference images have similar structures, they may fail when both images have totally irrelevant contents and styles.

Photorealistic image stylization Unlike color transfer, style transfer aims to modify a source image to follow the artistic style of a reference image, changing structural details. In addition to deep learning based style transfer methods [6, 19, 11], photorealistic style transfer methods have been proposed for changing the style of an image while suppressing structure distortions. Luan *et al.* [26] globally transfer the color style of the reference image using the Gram matrix while preserving original inter-pixel relationships using a matting Laplacian matrix. Li *et al.* [17] extend the idea of WCT [18] and propose PhotoWCT that replaces upsampling layers by location-preserving unpooling layers in the decoder. Yoo *et al.* [36] propose WCT^2 that uses wavelet unpooling layers instead of the location-preserving unpooling layers, enabling not only structural information but also statistical properties in the feature space to be preserved. Li *et al.* [16] train a deep network to learn a transformation matrix for color style transfer, and adopt spatial propagation network [23] to suppress structural distortions. While these methods have the benefit of transferring styles as well as colors in a photorealistic way, style transfer may inevitably incur distortions of structural details.

Palette based image recolorization There have been studies on color conversion using palettes. O'Donovan *et al.* [28] consider color compatibility to find globally preferred color combinations and recommend a fixed number of colors in the form of a palette. A follow-up study [22] automatically colorizes 2D graphic images using a probabilistic factor graph model. An interactive tool [3] was developed to provide image color manipulation through palette combinations. A deep learning method [4] has been proposed to transform the color of an image using a fixed sized palette. Compared to palettes, histograms contain still abstract but more information of image colors, and consequently could guide color transfer more effectively (Fig. 8).

3 Histogram-Driven Image Color Transfer

In this paper, we propose a feed-forward deep learning framework that leverages a reference color histogram to convert the color of a source image (Fig. 3). Since it is not straightforward for a deep network to learn color information from arbitrary images containing various spatial structures, we use histograms as an abstract but explicit clue for color transfer. Although the histogram itself is a global representation of color distribution, the network can learn spatially varying operations by leveraging the histograms as conditional information. We incorporate histogram analogy into our network by combining encoded source and reference histograms with the source image information to generate the color transfer result.

3.1 Framework Design

Our framework consists of two networks, histogram encoding network (HEN) and color transfer network (CTN). Our framework receives a source image (I_s), and source and reference histograms (H_s, H_t) for histogram analogy. Since a color histogram consists of a number of elements, it is difficult to train CTN to directly utilize the histogram information. Therefore, we adopt HEN to compress and encode the color histograms H_s and H_t into e_s and e_t. CTN receives a source image I_s as an input with two encoded histograms e_s and e_t, then it converts the colors of I_s. The encoded histograms are fused with intermediate feature maps of I_s in CTN (Sec. 3.3). To fuse spatial feature maps of I_s and 1D global feature vectors e_s and e_t, we tile the feature vectors to form histogram feature maps E_s and E_t with the spatial size of the corresponding layer to be concatenated. With the histogram feature maps, CTN can exploit the information of global histogram analogy in processing local regions.

Our framework may use optional semantic segmentation for more plausible color transfer, which we refer to as semantic replacement (Sec. 3.4). Previous works also explicitly or implicitly relate semantics between the source and reference images using semantic segmentation [26], dense semantic correspondence [21, 8], WCT using feature matching [18, 17, 36]. In our case, semantic replacement is an optional technique that can be used for explicitly correlating semantic regions between the source and reference images in the case of weak relevance.

We first compute local histograms from image regions with same semantic labels, and then feed the semantic-wise histograms into HEN to construct a spatially and semantically varying histogram feature map R_t instead of the global map E_t. With semantic replacement, CTN receives more precise local information of the reference histogram, which reduces uncertainty in determining colors so that challeng-

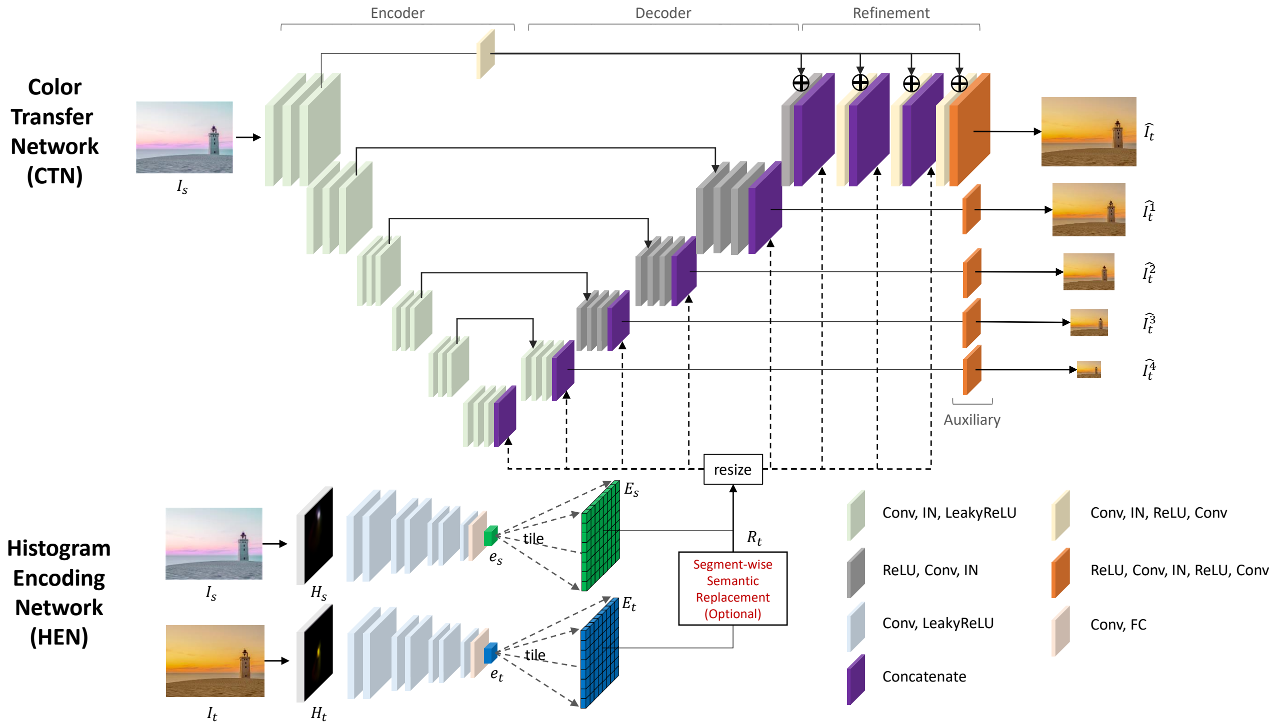

Fig. 3 Our network architecture. Our color transfer network (CTN) converts the color of the source image based on encoded source and reference histograms. Our histogram encoding network (HEN) takes a histogram as input and produces a feature vector of color distribution. The detail architecture can be found in Appendix A.

ing source and reference pairs with spatially different compositions can be handled.

3.2 Histogram Encoding Network (HEN)

Histogram encoding network (HEN) is a convolutional neural network that encodes the histogram of an image into a feature vector. Refer to HEN in Fig. 3 for the detailed architecture. The histogram H of an image, which is the input of HEN, is extracted by the following steps. We first divide an image I into I^L and I^{ab}, which contain the luminance and color information of the image in Lab color space. Then, we compute histograms $H^L \in \mathbb{R}^p$ of I^L and $H^{ab} \in \mathbb{R}^{q\times r}$ of I^{ab}, where p and $q \times r$ are histogram sizes. We tile H^L by $q \times r$ and concatenate it with H^{qr} in the channel direction to finally obtain $H \in \mathbb{R}^{q\times r\times(p+1)}$. HEN takes H and generates an encoded histogram $e \in \mathbb{R}^k$, where k is the histogram feature vector size. In our work, we use $q = 64$, $r = 64$, $p = 8$, and $k = 64$. There is no supervised loss to explicitly train HEN. Instead, HEN is trained together with CTN in an end-to-end manner using the loss functions defined in Sec. 3.3.

3.3 Color Transfer Network (CTN)

Our CTN consists of three modules: encoder-decoder, refinement, and auxiliary modules. For the encoder-decoder module, we adopt the structure of U-Net [13], which is used for various image processing problems. We use the ResNet [7] block for the refinement module to handle detailed color transformation. At the end of each level of the decoder, we attach an auxiliary layer to guide multi-scale prediction of the color transferred image. Refer to CTN in Fig. 3 for the detailed network architecture.

CTN takes a source image I_s as input, then the encoder embeds the image into a feature space. The encoded feature map of I_s is then fed into the decoder to produce a color transfer result. To guide color transfer, the encoded histogram information is repeatedly fused into features in the decoder at different scales and in the refinement module. Specifically, histogram feature maps $\{E_s, E_t\}$ (or $\{E_s, R_t\}$ for semantic replacement) obtained by tiling histogram feature vectors from HEN are resized and concatenated to the features at the end of each level of the decoder, and to the features in the refinement modules as shown in Fig. 3.

We perform supervised training by our paired dataset, as described in Sec. 4.1. During training, we jointly train HEN and CTN to minimize the following objective function:

$$\mathcal{L}_{total} = \mathcal{L}_{image} + \lambda_1\mathcal{L}_{hist} + \lambda_2\mathcal{L}_{multi}, \quad (1)$$

where $\mathcal{L}$, $\mathcal{L}_{hist}$ and $\mathcal{L}_{multi}$ are a image loss, a histogram loss, and a multi-scale loss, respectively. λ_1 and λ_2 are parameters to balance the terms.

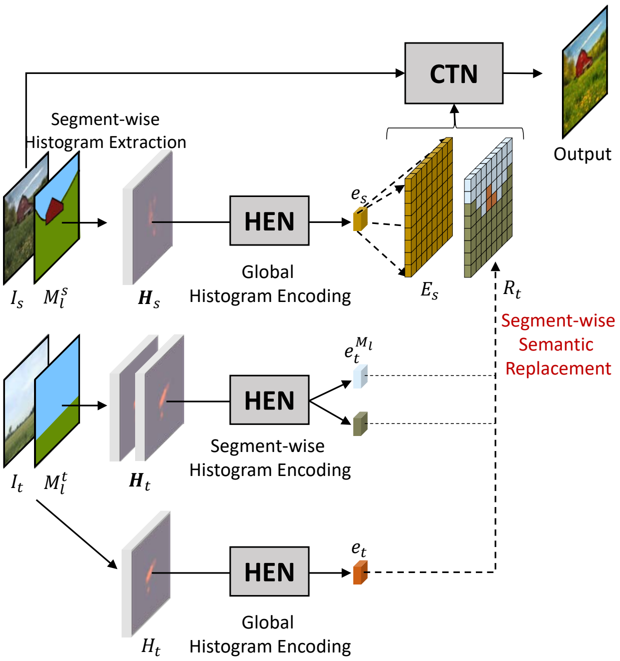

Fig. 4 The flow of semantic replacement while testing. The reference histogram is initialized by a globally encoded histogram of the reference image. Then, the reference histogram information is tiled differently according to the segmentation label of the source image.

Image loss The image loss $\mathcal{L}_{image}$ measures the difference between the color transfer result $\widehat{I}_t$ and the ground-truth image I_t, which is defined as

$$\mathcal{L}_{image} = MSE(\widehat{I}_t, I_t), \tag{2}$$

where *MSE* is the mean-squared-error between the inputs.

Histogram loss The histogram loss $\mathcal{L}_{hist}$ enforces the result image $\widehat{I}_t$ to have a similar histogram $\widehat{H}_t$ with the reference histogram H_t. To make the loss differentiable, we compute the histogram using convolution layers as in [34]. We define $\mathcal{L}_{hist}$ as follows:

$$\mathcal{L}_{hist} = MSE(\widehat{H}_t, H_t). \tag{3}$$

Multi-scale loss The role of each level of the decoder is to up-sample information from the previous level in the feature space. Our multi-scale loss $\mathcal{L}_{multi}$ explicitly enforces this role of each level of the decoder, and is defined by:

$$\mathcal{L}_{multi} = \frac{1}{D} \sum_{d=1}^{D} MSE(\widehat{I}_t^d, I_t^d), \tag{4}$$

where D is the number of up-sampling levels in the decoder.

3.4 Segment-wise Semantic Replacement

When the source and reference images are weakly relevant with high semantic similarity but low correlation in object configuration, globally applying histogram analogy would not always generate visually pleasing results. This is predictable as no clue of semantic relationship could be obtained by just looking at globally computed source and reference histograms. In that case, we explicitly provide additional semantic information to the network with *semantic replacement*.

The entire process is shown in Fig. 4. First, HEN produces encoded global histogram vectors e_s and e_t from H_s and H_t, respectively. Second, e_s and e_t are tiled to form histogram feature maps E_s and E_t to the sizes of the corresponding spatial feature maps. If source and target images are either strongly relevant or irrelevant, we feed E_s and E_t to CTN. Otherwise, we apply semantic replacement on E_t.

For semantic replacement, we first obtain semantic class-wise reference histogram feature vectors $e_t^{M_l}$ using HEN, based on the semantic segmentation result M^t of the reference image I_t. Specifically, for each semantic label l, we compute the histogram $H_t^{M_l}$ of I_t for regions with the same label l in M^t. Then, we feed each histogram to HEN to obtain $e_t^{M_l}$. We use semantic segmentation network [24] trained to predict $n = 8$ different labels (background, sky, water, grass, mountain, building, plant, animal and void) given an image.

Once E_t and $e_t^{M_l}$ have been obtained, we apply semantic replacement on E_t with $e_t^{M_l}$ using the source and reference segmentation maps M^s and M^t. For each matching semantic label l between M^s and M^t, we overwrite $e_t^{M_l}$ onto E_t at regions whose corresponding regions at the same locations in M^s have the label l. For regions on E_t with no matching semantic labels, no overwriting happens and the global histogram feature vector e_t remains in place. As a result, we obtain a semantic class-wise replaced histogram feature map R_t, and then CTN running with I_s, E_s, and R_t converts I_s according to histogram analogy combined with semantic information.

It can be argued that semantic replacement should also be applied to the source histogram feature map E_s. However, in our experiments, semantic replacement only on E_t produces better results. A possible reason is that replacing E_s as well as E_t would loose the globally coherent information on color transfer.

4 Experiments

4.1 Training

In color transfer, source and reference images are usually not the same, and it is not straightforward to train CNNs ef-

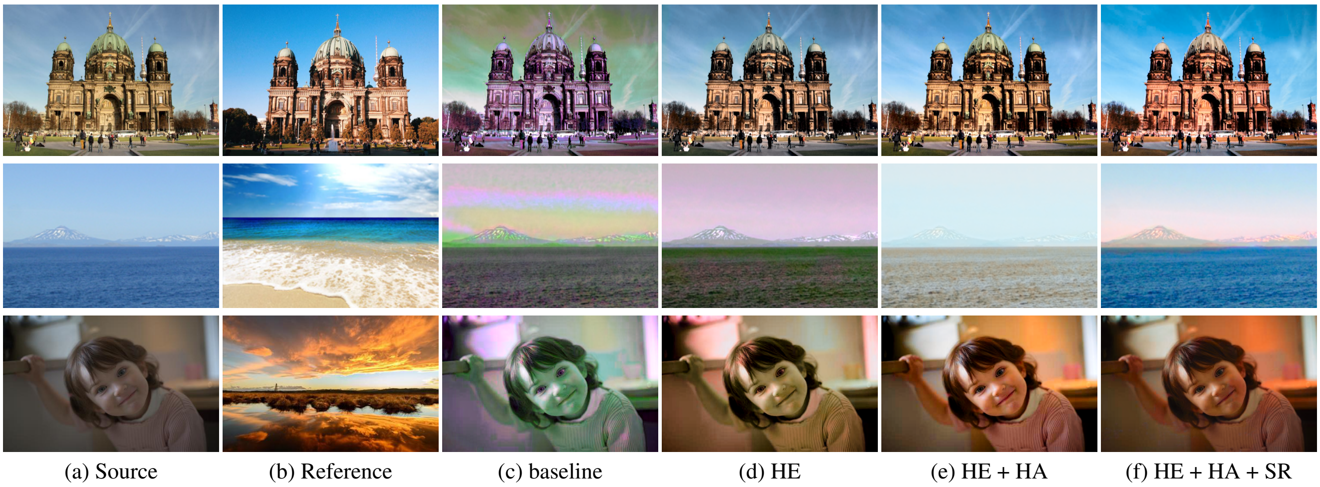

Fig. 5 Qualitative results of an ablation study. The first, second, and third rows show color transfer results of image pairs of strong relevance, weak relevance, and irrelevance, respectively.

ab bin size	8	32	64	128
PSNR(dB)	25.53	26.44	28.44	25.41

Table 2 Effectiveness of the bin size of the ab histogram.

Model	baseline	HE	HE + HA	HE + HA + SR
PSNR(dB)	10.71	26.27	27.92	28.44

Table 3 Quantitative results of an ablation study.

fectively with unpaired images. For supervised training, we use a dataset containing image pairs, which are structurally identical but have different colors. Although it seems that the pair dataset only covers strongly relevant cases, CTN can learn histogram analogy from the pair images because the network does not encounter the contents of reference images but only encoded histogram features. Moreover, semantic replacement enables the network to confront the dynamic distributions of semantic class-wise local histograms.

Dataset construction We use MIT-Adobe 5K dataset [2] for constructing a paired dataset, which consists of six sets, each of which contains 5,000 images. In the dataset, the first set contains the original images and the other five sets contain color varied images of the original images, retouched by five different experts, enabling our model to learn to transfer color into natural-looking images. Since using the dataset only provides image pairs with a fixed number of combinations, we perform color augmentation by transforming the average hue and saturation of the original images to produce more diverse image pairs, which enables the network to be well generalized to unseen images.

In addition, we use identical source and reference image pairs to stabilize the network output. An ideally trained model must produce the output image the same as the source image if the reference histogram is the same as the source histogram itself. Using identical image pairs enforces this constraint and improves training in our preliminary test. Half of our training dataset consists of identical image pairs.

Implementation details We implement our network on Pytorch framework and used an NVIDIA Titan XP GPU to train the network. For training, we use the Adam optimizer [12] with the fixed learning rate $5 * 10^{-5}$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. We use $\lambda_1 = 1.5$ and $\lambda_2 = 0.5$ in Eq. (1). It takes about 100 epochs for out network to converge. CTN and HEN are jointly trained with semantic replacement.

We adopt $p = 64$ and $r = 64$ for the model used for the rest of experiments according to Table 2, which shows the performance of the final model with varying bin size of the ab histogram. We experimented with the ab bin size as it is the key factor in terms of color transfer.

4.2 Ablation Study

In this section, we show the effectiveness of our proposed components, which are histogram encoding (HE), histogram analogy (HA), and semantic replacement (SR).

Table 3 shows quantitative comparison results of four variations of our models. The first model is a baseline model where HEN receives a reference image while CTN receives a source image as well as encoded image features from HEN. In the second model (HE), we feed a reference histogram to HEN and CTN receives a source image and an encoded histogram from HEN. In the third model (HE + HA), we supply an encoded source histogram to the second model. The final model (HE + HA + SR), we apply semantic replacement in the third model. For quantitative comparisons, we make 500 test image pairs, where the content of each image is the same but with different colors, and measure PSNRs between the results of the models and their corresponding GT images.

Fig. 5 shows visual results on three cases of image pairs: strong, weak relevance, and irrelevance. The baseline model fails to reflect the reference colors and it tends to produce consistent color tones. The HE model produces an image with vivid colors, but it also fails to reflect the reference colors properly. The HE + HA model, on the other hand, produces well transferred results in the strong relevance and irrelevance cases. However, it shows an unsatisfactory result in the weak relevance case. Our final model (HE + HA + SR) shows plausible results in all three cases. It shows that our histogram analogy and semantic replacement effectively train the framework of our histogram-driven image color transfer.

4.3 Qualitative evaluation

We compare our method with two GCT methods (Reinhard *et al.* [30] and Pitie *et al.* [29]), one LCT method (He *et al.* [8]), and four style-transfer based LCT methods (Liao *et al.* [21], Luan *et al.* [26], Li *et al.* [17], and Yoo *et al.* [36]). For comparison, we use implementations provided by the authors. Among the methods, Luan *et al.*'s method requires semantic segmentation maps regardless of the relevance between source and reference images. On the other hand, Yoo *et al.*'s and Li *et al.*'s methods can optionally use semantic segmentation maps. Thus, for Yoo *et al.*'s and Li *et al.*'s methods in our experiments, we use semantic segmentation maps only for weakly relevant image pairs as done for ours. We compare the methods in three different categories: strongly relevant, weakly relevant, and irrelevant cases (Fig. 6).

Strongly relevant case While most methods work well for the strongly relevant case, some methods tend to fail to preserve structures, when images have dense textures (Fig. 6e,g), Our proposed method generally works well for strongly relevant image pairs.

Weakly relevant case GCT methods [30,29] tend to fail in the case of weak relevance (Figs. 6c and d). LCT based methods usually well reflect the reference colors, some methods produce results with unnatural artifacts (Figs. 6e,f,g,i). He *et al.*'s method [17] show a preferable result in term of color reflection, but artifacts can be found in their result (Fig. 6h). Our method (Fig. 6j) shows the best result with moderate color reflection as well as content preservation.

Irrelevant case Style transfer based LCT methods usually fail to preserve contents (Fig. 6e,f,g). He *et al.*'s method (Fig. 6h) using dense correspondences reflects the reference colors well, but produces exaggerated textures. As shown in Fig. 6c and d, GCT methods [30,29] preserve texture information, while Pitie *et al.*'s method [29] makes unnatural colors. GCT based methods, Yoo *et al.*, and our method (Fig. 6c,d,i,j) produce reasonable results without noticeable artifacts.

A summary of the qualitative evaluation is that our method shows moderate performance for all three cases while other methods fail to color transfer in more than one case. We refer the readers to our supplementary material for more results.

4.4 User Study

For further evaluation, we conduct a user study using Amazon Mechanical Turk (AMT) with 20 participants and 30 test images. We compare our results with those of five previous methods, Pitie *et al.* [29], Luan[26], He *et al.* [8], Yoo *et al.* [36]. For the mehtod of Luan *et al.*, we set 1,000 iterations for the optimization. The benchmark dataset consists of image pairs provided by Lee *et al.* [15] and Luan *et al.* [26]. Total 300 questions are asked to 20 subjects who are asked to choose one image between a pair of images. Each question consists of three sub-questions that ask the subjects to choose preferred images in terms of color reflectance, contents preservation, and overall quality, respectively. After we collect total 18,000 pairwise comparison results, we compute a preference matrix. As shown in Fig. 7, which is summarization of the preference matrix, our method is preferred than the other methods in all criteria except one, where He *et al.*'s method [8] is favored for color reflectance in the strong relevance case. More details, raw data and the survey samples can be found in Appendix B.

5 Applications

Since color histograms can be extracted from various inputs and edited easily, our method using histogram analogy can be adopted in various applications, including palette-based recolorization, photo editing with histogram modification, and color enhancement via histogram stretching. Furthermore, our framework does not need additional training to support these applications. In this section, we demonstrate palette-based recolorization, color enhancement via histogram stretching, and photo editing with histogram modification.

5.1 Palette-based recolorization

Palette-based recolorization allows users to easily transform image colors by choosing several colors in a palette without effort to find specific reference images. However, a color palette is a kind of severely irrelevant reference image, so conventional color transfer methods for general purpose may not work well with color palettes (Fig. 8b). Also, palette-based recolorization methods force result images to have specific palette colors, as they use the palette colors as a hard constraint. As a result, they could produce less plausible results (Fig. 8c). On the other hand, our method receives palette colors as a guidance, and produces naturally recolorized results (Fig. 8d).

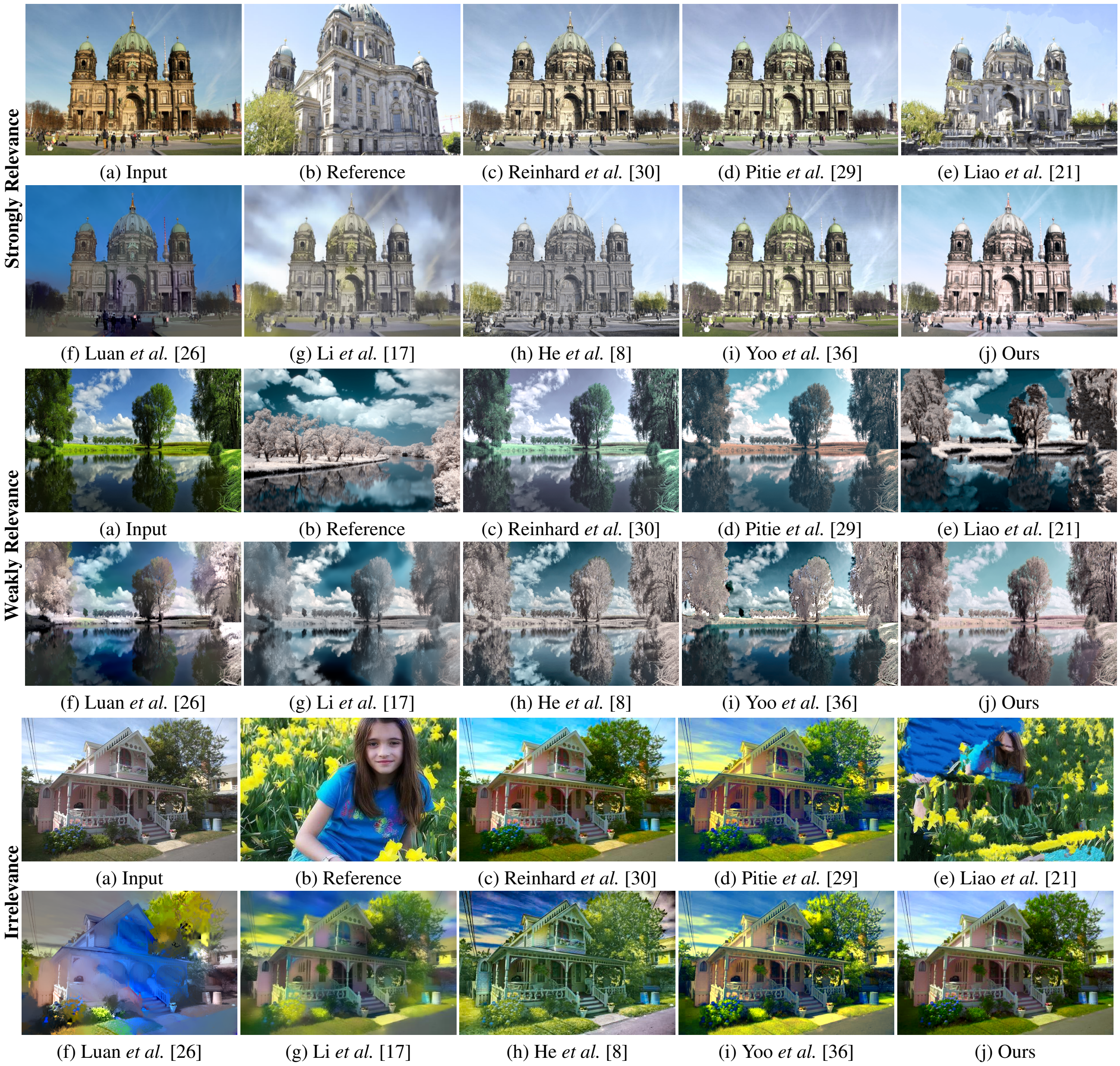

Fig. 6 Color transfer results on various source and reference image pairs. For visualization, the reference image is cropped to make a same size with other images.

5.2 Color Enhancement via Histogram Stretching

In this application, our method takes a stretched histogram of an input source image as a reference histogram, without the need of a reference image. For histogram stretching, we *spatially* enlarge the source histogram in the Lab color space with respect to the center of the 2D histogram of I_s^{ab} and normalize the summation of values of all bins to 1. As a stretched histogram contains a wider range of colors, our method produces natural-looking images with more vivid colors (Fig. 9).

5.3 Photo Editing with Histogram Modification

Our method can be used for histogram-based photo editing, where a user modifies an image by manipulating its histogram. Fig. 10 shows such an example. In the example, red is replaced by purple in the histogram. Then, our method produces a color transfer result that is natural-looking and consistent with the manipulated histogram.

6 Discussions and Limitations

We propose a novel feed-forward network for image color transfer that leverages deep encoded histogram features. While

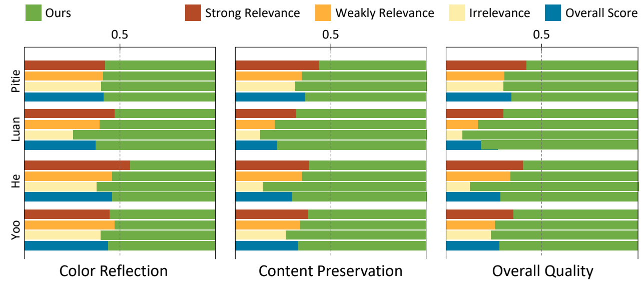

Fig. 7 User study result by amazon mechanical turk (AMT). Each participants were asked to answer three aspects: contents preservation, color reflection and preference. Our method achieved the best performance in terms of total averaged score.

(a) input (b) Pitie *et al.* [29] (c) Chang *et al.* [3] (d) Ours

Fig. 8 Palette-based recolorization results. At each input image, the reference palette colors are included at the top-left corner.

Input (× 1.0) × 1.2 × 1.4 × 1.8

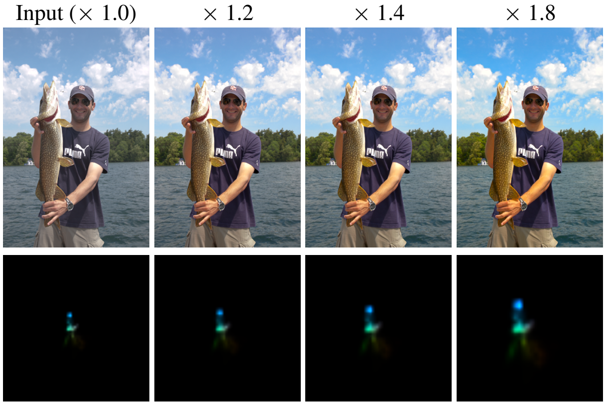

Fig. 9 Color enhancement by histogram stretching. The images in the first column are the source image and its histogram. The other columns demonstrate that we can obtain images with enhanced colors (upper row) using stretched histograms as the references (bottom row).

Fig. 10 Photo editing with histogram modification. Images in the first column are the input (upper) and its histogram (bottom). Images in the second column are the output (upper) generated with the modified histogram (bottom) as its reference.

Source Reference Our results

Fig. 11 Failure examples using severely erroneous segmentation maps.

previous methods are not capable of covering diverse pairs of source and reference images, our network effectively converts a color of a source image according to diverse reference images based on deep histogram analogy. With semantic segmentation information, our method can also handle image pairs with different compositions, but semantically similar contents. For validating the capability of our method to deal with diverse scenes, we divided diverse cases into three categories by a degree of image correlation; *strong relevance*, *weak relevance*, and *irrelevance*. Extensive experiments show that our method shows moderate performance for all the cases, comparable to the state-of-the-art color transfer methods specialized for a part of the items.

Although image-to-image translation seems related to color transfer, it focuses more on learning inter-domain translations between different domains while color transfer is rather about intra-domain translation. We tested GAN-based image-to-image translation methods, such as DRIT [14] and GDWCT [35], for color transfer. In our experiments, only DRIT reasonably worked but still produced low-quality results, compared to existing color transfer methods.

Limitation As our semantic replacement depends on semantic segmentation, the performance of our method with semantic replacement can be degraded by segmentation error (Fig. 11). In such a case, a user may simply turn off semantic replacement as our method can still provide natural-looking results without semantic replacement. More advanced semantic segmentation techniques may improve the performance of our method.

Acknowledgements

This work was supported by the Ministry of Science and ICT, Korea, through IITP grant (IITP-2015-0-00174) and NRF grants (NRF-2017M3C4A7066316, NRF-2017M3C4A7066317).

References

1. Arbelot, B., Vergne, R., Hurtut, T., Thollot, J.: Local texturebased color transfer and colorization. In: Computers & Graphics 62 (2017)
2. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input / output image pairs. In: The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition (2011)
3. Chang, H., Fried, O., Liu, Y., DiVerdi, S., Finkelstein, A.: Palette-based photo recoloring. In: ACM Transactions on Graphics (2015)
4. Cho, J., Yun, S., Lee, K., Choi, J.: Palettenet: Image recolorization with given color palette. In: CVPR Workshops (2017)
5. Freedman, D., Kisilev, P.: Object-to-object color transfer: Optimal flows and smsp transformations. In: Proc. CVPR (2010)
6. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. In: CoRR (2015)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. CVPR (2016)
8. He, M., Liao, J., Chen, D., Yuan, L., Sander, P.V.: Progressive color transfer with dense semantic correspondences. ACM Trans. Graph. **38**(2), 13:1–13:18 (2019)
9. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proc. SIGGRAPH (2001)
10. Hristova, H., Meur, O.L., Cozot, R., , Bouatouch, K.: Style-aware robust color transfer. In: Proc. workshop on Computational Aesthetics (2015)
11. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proc. ICCV (2017)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. ICLR (2014)
13. Kingma, D.P., Ba, J.: U-net: Convolutional networks for biomedical image segmentation. In: In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015)
14. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M.K., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: Proc. ECCV (2018)
15. Lee, J., Sunkavalli, K., Lin, Z., Shen, X., Kweon, I.: Automatic content-aware color and tone stylization. In: Proc. CVPR (2016)
16. Li, X., Liu, S., Kautz, J., Yang, M.H.: Learning linear transformations for fast arbitrary style transfer. In: Proc. CVPR (2019)
17. Li, Y.: A closed-form solution to photorealistic image stylization. In: Proc. ECCV (2018)
18. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.: Universal style transfer via feature transforms (2017)
19. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. In: Proc. NIPS (2017)
20. Li Cheng, Vishwanathan, S.V.N., Xinhua Zhang: Consistent image analogies using semi-supervised learning. In: Proc. CVPR (2008)
21. Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. In: Proc. SIGGRAPH (2017)
22. Lin, S., Ritchie, D., Fisher, M., Hanrahan, P.: Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. In: ACM Transactions on Graphics (2013)
23. Liu, S., Mello, S.D., Gu, J., Zhong, G., Yang, M., Kautz, J.: Learning affinity via spatial propagation networks. In: Proc. NIPS (2017)
24. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Deep learning Markov random field for semantic segmentation. In: TPAMI (2017)
25. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proc. CVPR (2015)
26. Luan, F., Paris, S., E.Shechtman, Bala, K.: Deep photo style transfer. In: Proc. CVPR (2017)
27. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. In: https://distill.pub/2016/deconv-checkerboard/ (2016)
28. O'Donovan, P., Agarwala, A., Hertzmann, A.: Color compatibility from large datasets. In: Proc. SIGGRAPH (2011)
29. Pitie, F., Kokaram, A., Dahyot, R.: N-dimensional probability density function transfer and its application to color transfer. In: Proc. ICCV (2005)
30. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. In: IEEE Computer graphics and applications 21(5) (2001)
31. Tai, Y.W., Jia, J., , Tang, C.K.: Local color transfer via probabilistic segmentation by expectation-maximization. In: Proc. CVPR (2005)
32. Tsai, Y.H., Shen, X., Lin, Z., Sunkavalli, K., Yang, M.H.: Sky is not the limit: Semantic-aware sky replacement. In: ACM Transactions on Graphics (2016)
33. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. In: arXiv preprint arXiv:1607.08022 (2016)
34. Wang, Z., Li, H., Ouyang, W., Wang, X.: Learnable histogram: Statistical context features for deep neural networks (2016)
35. Wonwoong Cho Sungha Choi, D.K.P.I.S.J.C.: Image-to-image translation via group-wise deep whitening-and-coloring transformation. In: Proc. CVPR (2019)
36. Yoo, J., Uh, Y., Chun, S., Kang, B., Ha, J.: Photorealistic style transfer via wavelet transforms. In: Proc. ICCV (2019)

Appendix

A Network Architecture Details

Color Transfer Network (CTN) Our proposed CTN adopts modified structures of U-Net [13] and ResNet [7]. CTN consists of four submodules: encoder, decoder, residual blocks, and auxiliary modules[1].

The encoder consists of five convolutional blocks, where each block consists of a varying number of convolutional layers followed by instance normalization [33] and leaky ReLU layers. The encoder and decoder are connected through skip connections, and the features produced by the encoder are passed to the decoder. The decoder also consists of five convolution blocks. To avoid checkerboard artifact [27], each convolutional block uses upsampling convolution, which performs convolution after bilinear upsampling, instead of using deconvolution. Then, the final feature map of the decoder goes into the residual blocks, which produce the final recolored image. In addition, for training stability, we attach auxiliary modules at the end of each convolutional block in the decoder to produce temporary result images of smaller scales.

Histogram Encoding Network (HEN) HEN consists of eight convolution layers, each of which is followed by leaky ReLU activation. Given input and reference images, HEN generates an encoded histogram feature for each of input and reference images.

B User Study Details

We conducted the user study using Amazon Mechanical Turk (AMT) with 20 participants whose hit approval rates are over 98%, and 30 source/ reference image pairs from [15,26]. The image pairs cover all three cases: strong relevance, weak relevance, and irrelevance, each of which has 10 pairs.

[1] Readers may also refer to the code that will be released.

		Color Reflection						Content Preservation						Overall Quality					
		[29]	[26]	[8]	[36]	Ours	total	[29]	[26]	[8]	[36]	Ours	total	[29]	[26]	[8]	[36]	Ours	total
Strong	[29]	-	104	91	104	85	384	-	109	118	111	88	426	-	120	108	118	84	430
	[26]	96	-	94	113	95	398	91	-	85	83	64	323	80	-	80	88	60	308
	[8]	109	106	-	127	111	**453**	82	115	-	98	78	373	92	120	-	107	80	399
	[36]	96	87	73	-	90	346	89	117	102	-	77	385	82	112	93	-	70	357
	Ours	115	105	89	110	-	419	112	136	122	123	-	**493**	116	140	120	130	-	**506**
Weak	[29]	-	120	103	117	83	423	-	161	105	124	70	460	-	160	110	127	61	458
	[26]	80	-	96	86	79	403	39	-	57	63	42	201	40	-	56	62	33	191
	[8]	97	104	-	110	92	341	95	143	-	115	71	424	90	144	-	123	67	424
	[36]	83	114	90	-	95	382	76	137	85	-	69	367	73	138	77	-	51	339
	Ours	117	121	108	105	-	**451**	130	158	129	131	-	**548**	139	167	133	149	-	**588**
Irrelevant	[29]	-	142	130	112	81	465	-	165	156	126	63	510	-	176	159	133	60	528
	[26]	58	-	75	58	52	243	35	-	68	63	27	193	24	-	64	53	17	158
	[8]	70	125	-	68	76	399	44	132	-	59	30	265	41	136	-	62	24	263
	[36]	88	142	132	-	80	442	74	137	141	-	54	406	67	147	138	-	47	399
	Ours	119	148	124	120	-	**511**	137	173	170	146	-	**626**	140	183	176	153	-	**652**

Table 4 The preference matrix from the user study of strong relevance pairs.

Fig. 12 User study samples. For visualization, the reference image is cropped to make a same size with other images.

We compared our method with four state-of-the-art color transfer methods: Pitie *et al.* [29], Luan *et al.* [26], He *et al.* [8], and Yoo *et al.* [36], in a pairwise manner. Specifically, we showed each participant all possible pairs of the results of different color transfer methods side-by-side as well as their original and reference images. As a result, each participant was shown 300 pairs of result images generated by five methods. Then, for each result image pair, we asked the participants three questions: 1) which image better preserves the content of the original image? 2) which image better reflects the color of the reference? and 3) which image is better in terms of overall quality?

Table 4 show the user study result in the form of preference matrices. Each element in the matrices shows the number of votes for one method (specified by its row) over another method (specified by its column). For instance, the element at the fifth row and second column of strong relevant case in Table 4 means that our results are preferred over the results of Luan *et al.* [26] by 68% (136 among 200 votes) in the aspect of content preservation in the case of strong relevance. As the tables illustrate, our method outperforms the others in terms of color reflection, content preservation and overall quality. Figs. 12 and 13 show some samples used in the survey.

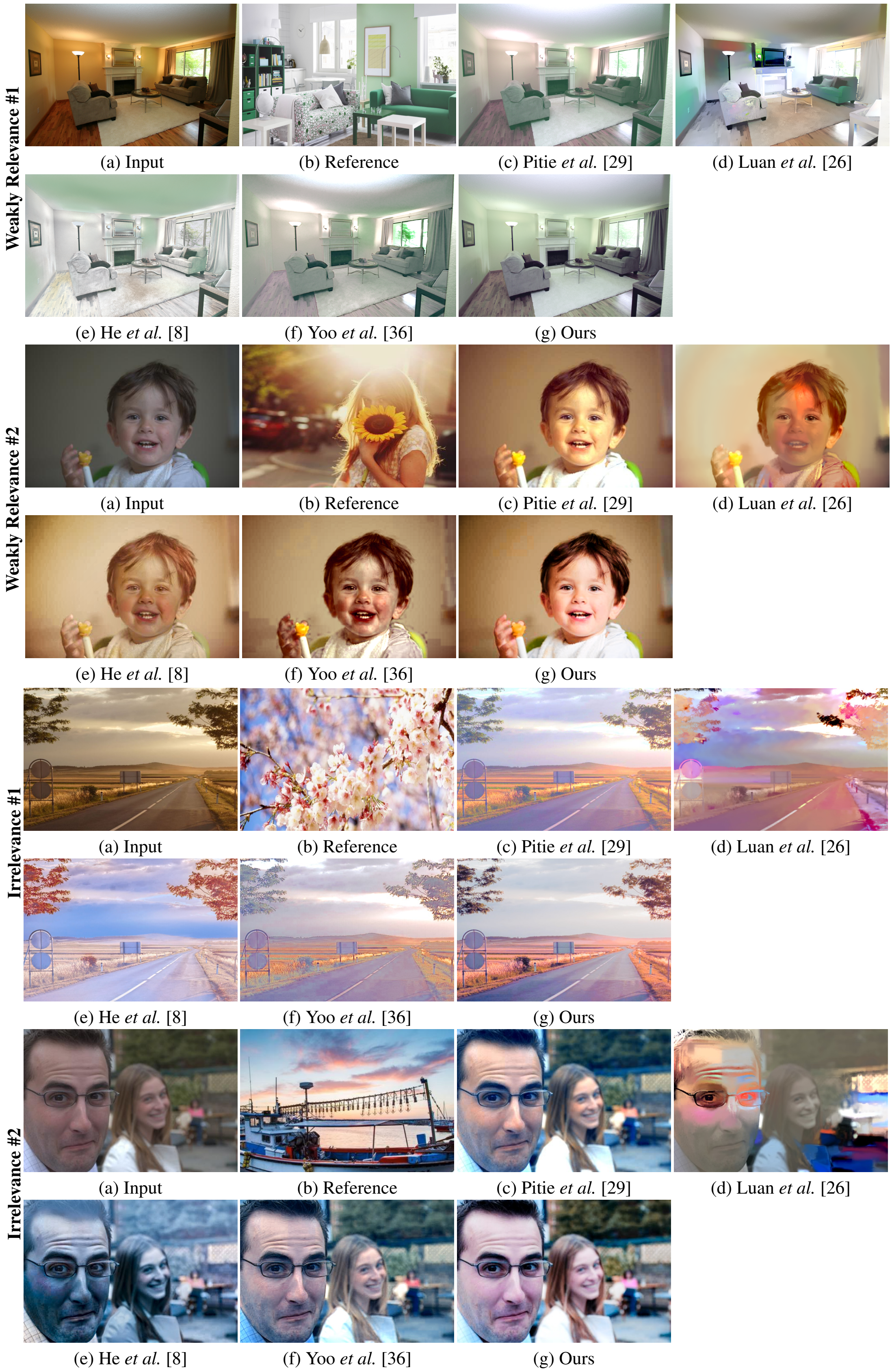

Fig. 13 User study samples. For visualization, the reference image is cropped to make a same size with other images.