Invertible Grayscale

MENGHAN XIA, Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, SIAT, China and The Chinese University of Hong Kong
XUETING LIU, The Chinese University of Hong Kong
TIEN-TSIN WONG*, The Chinese University of Hong Kong and Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, SIAT, China

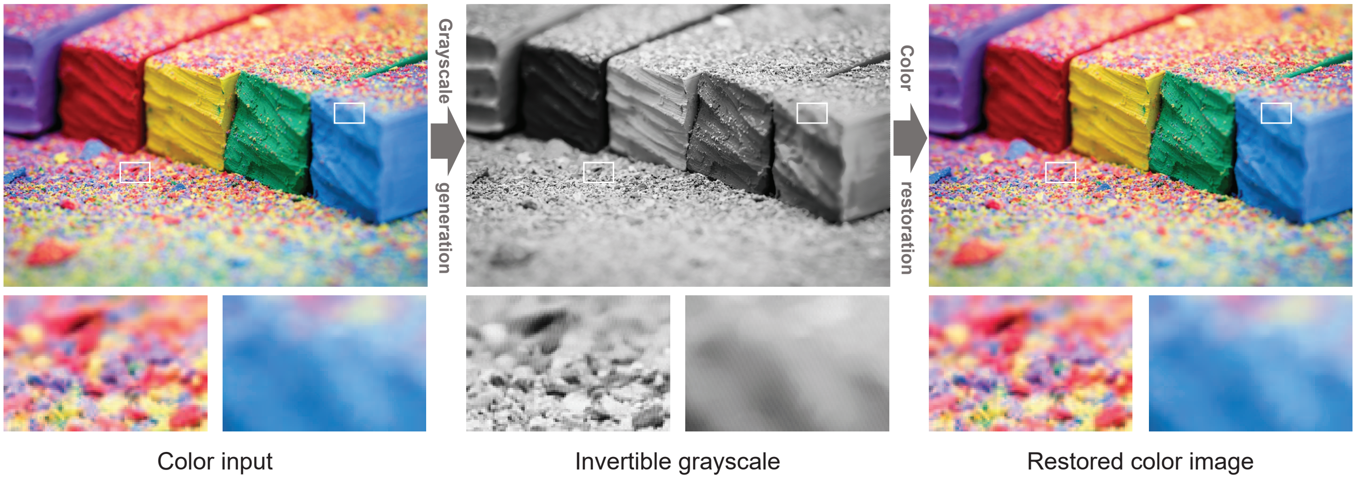


Fig. 1. Given an input color image (left), our method converts it to an invertible grayscale (middle) that can be later restored to the color version (right). The key of our method is to encode the original color information in the generated grayscale as unobvious pattern (blown-ups of the middle image) via a convolutional neural network. (MSE:2.627 PSNR: 36.76, SSIM: 0.9622)

Once a color image is converted to grayscale, it is a common belief that the original color cannot be fully restored, even with the state-of-the-art colorization methods. In this paper, we propose an innovative method to synthesize *invertible grayscale*. It is a grayscale image that can fully restore its original color. The key idea here is to encode the original color information into the synthesized grayscale, in a way that users cannot recognize any anomalies. We propose to learn and embed the color-encoding scheme via a convolutional neural network (CNN). It consists of an encoding network to convert a color image to grayscale, and a decoding network to invert the grayscale to color. We then design a loss function to ensure the trained network possesses three required properties: (a) color invertibility, (b) grayscale conformity, and (c) resistance to quantization error. We have conducted intensive quantitative experiments and user studies over a large amount of color images to validate the proposed method. Regardless of the genre and content of the color input, convincing results are obtained in all cases.

CCS Concepts: • **Computing methodology** → **Computational photography**;

Additional Key Words and Phrases: color encoding, invertible grayscale, image reconstruction

ACM Reference Format:
Menghan XIA, Xueting LIU, and Tien-Tsin WONG. 2018. Invertible Grayscale. *ACM Trans. Graph.* 37, 6, Article 246 (November 2018), 10 pages. https://doi.org/10.1145/3272127.3275080

1 INTRODUCTION

People convert color images to grayscale for various purposes and applications, ranging from black-and-white printing, aesthetic photography, to backward compatibility for legacy display. However, color information is lost during the color-to-gray conversion. Although ones can colorize grayscale using existing colorization methods [Iizuka et al. 2016; Zhang et al. 2016], the introduced colors may not be the original ones.

In this paper, we propose an innovative method to convert a color image to grayscale, that can be later inverted back to its color version. We call it *invertible grayscale*. Fig. 1 shows the invertible grayscale (middle) generated from the color input (left), while the right image is the restored color image. The key idea here is to encode the original color information in the generated grayscale,

[*] Corresponding author

Authors' addresses: Menghan XIA, Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, SIAT, Shenzhen, China, The Chinese University of Hong Kong, Hong Kong, mhxia@cse.cuhk.edu.hk; Xueting LIU, The Chinese University of Hong Kong, Hong Kong, xtliu@cse.cuhk.edu.hk; Tien-Tsin WONG, The Chinese University of Hong Kong, Hong Kong, Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, SIAT, Shenzhen, China, ttwong@cse.cuhk.edu.hk.

0730-0301/2018/11-ART246 $15.00
https://doi.org/10.1145/3272127.3275080

so that the encoded color information can be restored during the later color restoration. The blown-up grayscale sub-images on the bottom row of Fig. 1 reveal how the color is encoded as unobvious pattern. However, designing such a color encoding scheme by hand is extremely complex due to the unbounded content of input.

Instead, we propose to learn and embed the color encoding scheme via a convolutional neural network (CNN). Our system consists of an encoding neural network to convert a color image to grayscale, and a decoding neural network to invert the grayscale to color. A key to utilize a convolutional neural network lies on the design of a proper loss function. Our loss function is composed of invertibility loss, grayscale conformity loss, and quantization loss. The invertibility loss ensures the restored and the groundtruth color images to be as similar as possible, so that we can restore the color image. The grayscale conformity loss ensures that the generated grayscale looks like the grayscale version of the color input. This requires the generated grayscale to preserve the structure, contrast, and lightness of the color input. The quantization loss mimics the quantization process when we store the invertible grayscale in 8-bit per pixel form.

Once the color encoding and restoration scheme is trained, we can apply it to arbitrary color images without any restriction on the visual content. To validate our model, we tested it over 3, 000 natural images of different genre and content. Qualitative and quantitative evaluations, as well as user study, are conducted. Convincing statistics is obtained in all experiments. For instance, when comparing the restored color images to the original input, we achieve very high average SSIM of 0.9681 and average PSNR of 36.02 dB. Our contributions can be summarized as follows:

- We propose an innovative method to convert a color image to grayscale, that can be later inverted back to its color version, whenever necessary.
- We propose to formulate the color encoding scheme in a neural network framework, so as to provide an effective and efficient solution to generate invertible grayscale.

While our current method is tailored for the invertible color-to-gray problem, the general neural network framework proposed can be applied to many other applications that require an inversion ability. The information to be encoded in the resultant images can be color, or any other information. The framework serves to encode the information in a visually unobvious manner while the encoded information is restorable through the decoding network.

2 RELATED WORKS

2.1 Decolorization

Decolorization methods convert color images to grayscale. Note that there is no absolute standard for creating grayscale, because the visible light is a spectral quantity which is always colorful in the real world. Different applications utilize different criteria or standards in generating grayscale. Early grayscale generation serves the purpose of mimicking early monochrome photography that is unable to capture and present colors. The simplest and most commonly used color-to-gray method is to regard the lightness/luminance channel (either in CIELab color space or YUV/YIQ color space) of the color image as the gray value in the target grayscale image. However, this naive approach not only leads to color information loss, but sometimes also causes contrast or structural information loss when neighboring regions have similar gray values. To resolve this problem, various decolorization methods have been proposed to preserve the color contrast during the color-to-gray conversion. They can be classified into two types: local decolorization and global decolorization.

The local decolorization methods enhance the contrast in grayscale based on local chrominance edges. Bala et al. [2004] proposed to add high-frequency chromatic components to the lightness channel for preserving the chrominance edge information. Neumann et al. [2007] proposed a consistent gradient field notion based on local color and luminance contrast. Lu et al. [2012] proposed to find suitable color orders with respect to the visual context by optimizing a bimodal distribution. The global decolorization methods intend to preserve both local and non-local contrast via a global color mapping [Gooch et al. 2005; Kim et al. 2009; Kuk et al. 2010; Song et al. 2010]. Gooch et al. [2005] proposed to optimize a mapping rule based on a contrast reference map via a linear model. Kuk et al. [2010] extended Gooch's work by taking both local and global contrast into account. Liu et al. [2015] proposed to compute the gradient correlation between the color input and the targeted grayscale image. Although our method also tries to preserve the contrast during the invertible grayscale generation, none of the above methods aims at generating grayscale images that can be restored to color version, as we do.

2.2 Colorization

Colorization is a classic problem that aims at colorizing grayscale images/videos, especially legacy photographs and footages. Due to the color information loss, user-hints, such as user-scribbles or user-selected reference images, are typically needed during the process. The colorization process can be formulated as an optimization by propagating user-guided colors based on low-level features [Huang et al. 2005; Levin et al. 2004]. To increase the reliability of low-level features, more advanced similarity metrics [Luan et al. 2007; Qu et al. 2006] and long-range connection [An and Pellacini 2008; Xu et al. 2009] have also been explored. Global color consistency can be achieved by enforcing the global color theme [Wang et al. 2010] or color palette [Chang et al. 2015]. However, these methods require extensive manual intervention to achieve good colorization results. User-provided reference images can also be used for guiding the colorization [Charpiat et al. 2008; Chia et al. 2011; Irony et al. 2005; Liu et al. 2008; Welsh et al. 2002]. However, reference images must be carefully selected in order to obtain good results. This is also time-consuming and sometimes similar-content reference image may not be available.

Instead of providing user-hints, recently, deep-learning approaches are explored to learn the correlation of color and textures/objects, so that they can automatically colorize grayscale images with high and stable performance [Cheng et al. 2015; Deshpande et al. 2015; Larsson et al. 2016; Zhang et al. 2016]. However, these methods may not work well for images without sufficient textures or obvious objects. In order for users to control or provide hints, more CNN-based colorization methods have been proposed to accept user-interaction [Sangkloy et al. 2017; Zhang et al. 2017]. Tailored

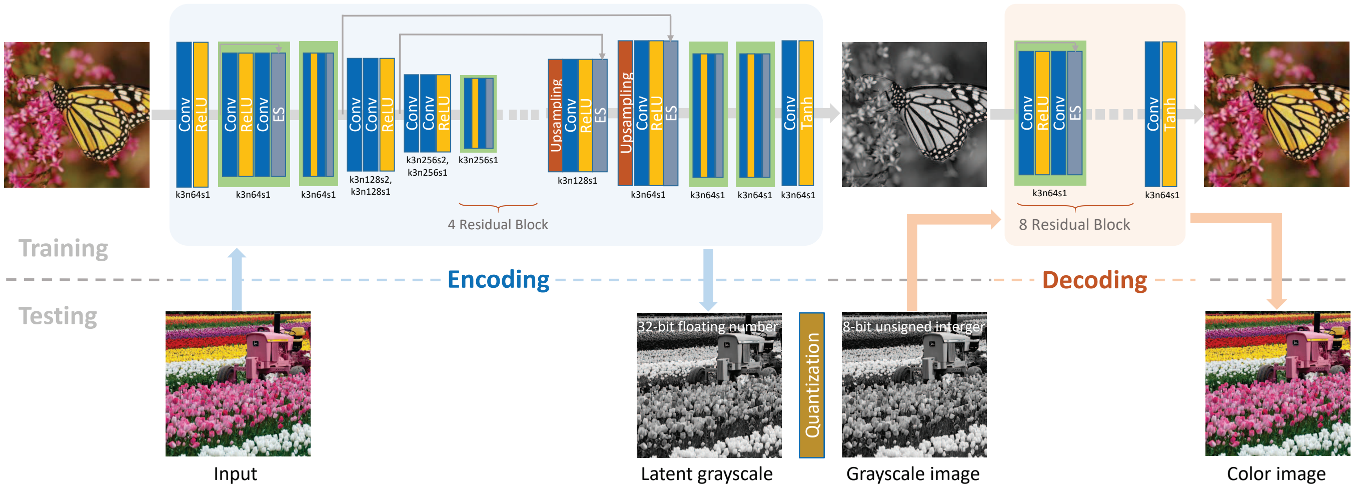


Fig. 2. System overview and network architectures. In both encoding and decoding networks, k denotes the kernel size. n denotes the number of feature maps. s denotes the stride in each convolutional layer. "ES" indicates elementwise addition.

for image compression, Baig and Torresani [2017] proposed to store the approximate color information of an image as meta-data, which can be further used for restoring the color image. Nevertheless, none of the existing colorization methods can guarantee to recover the original colors. In sharp comparison, our invertible grayscale encodes the color information so that the restoration of the original colors becomes feasible. Note that, although we also utilize a CNN in our method, it serves the color encoding purpose that is completely different from existing colorization works.

3 OVERVIEW

Our system is overviewed in Fig. 2, which consists of an encoding neural network E and a decoding neural network D. The encoding neural network E converts any color image in RGB color space to a grayscale image. Conversely, the decoding neural network D converts any grayscale image to a color image in RGB space. The network architecture of E and D is detailed in Section 4.

In the training phase, for each color image I in the training dataset, we convert it into a grayscale image G with the encoder E, and then restore a color image R from G with the decoder D. That is,

$$G = E(I) \tag{1}$$

$$R = D(G) = D(E(I)). \tag{2}$$

By laying requirements on both the grayscale image G and the restored color image R, and hence the encoder E and the decoder D, we jointly train E and D. In particular, we design three losses: invertibility loss $\mathcal{L}_V(E, D)$ that ensures I and R to be similar, grayscale conformity loss $\mathcal{L}_C(E)$ that ensures G to look like a grayscale image and conform to I, and quantization loss $\mathcal{L}_Q(E)$ that encourages the pixel values of G to be integers. The overall loss function $\mathcal{L}(E, D)$ is then formulated as a weighted sum of these three losses:

$$\mathcal{L}(E, D) = \mathcal{L}_V(E, D) + \omega_1 \mathcal{L}_C(E) + \omega_2 \mathcal{L}_Q(E) \tag{3}$$

The detailed design of the losses will be discussed in Section 5. The training scheme is detailed in Section 6. The weighting parameters are also discussed in Section 6 since the weights are different in different training stages.

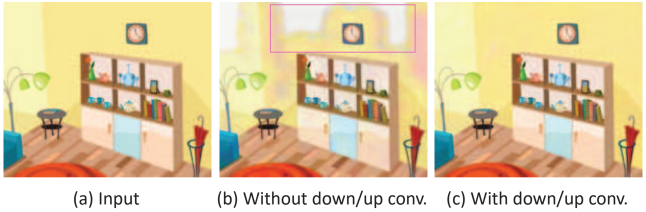

(a) Input (b) Without down/up conv. (c) With down/up conv.

Fig. 3. (a) Input color image. (b) Without using down/up convolution structure, significant artifacts can be observed on the wall in the restored color image. (c) With down/up convolution structure, the restored color image successfully maintains the appearance of the input image.

In the testing phase, given any color input I, we first convert it into a grayscale image $G = E(I)$. Note that each pixel of G is actually a 32-bit floating-point value. So we have to quantize G to an 8-bit grayscale image $\overline{G}$, so that all pixel values in $\overline{G}$ are integers in $[0, 255]$. The restored color image R can be obtained via $R = D(\overline{G})$. Results and indepth validation experiments are presented in Section 7.

4 NETWORK ARCHITECTURE

As mentioned above, our system consists of an encoding network converting color input to grayscale, and a decoding network converting grayscale to color. At first glance, it looks like a cyclic conversion between color and grayscale images, and seems to be solveable using CycleGAN [Zhu et al. 2017]. However, CycleGAN only guarantees that the output images conform to the corresponding image classes (either grayscale or color). Due to the unsupervised nature of CycleGAN, there is no guarantee that the generated grayscale conforms to the corresponding input color image as required in our case. Our encoding-and-decoding framework is also close to the hash autoencoder [Carreira-Perpinán and Raziperchikolaei 2015; En et al. 2017]. However, though the existing autoencoders may achieve invertibility, none of them imposes any requirement on the visual appearance of the latent representation.

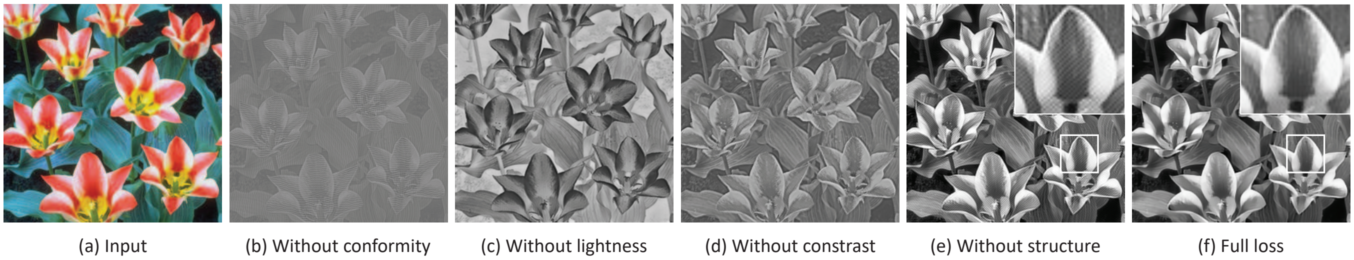


Fig. 4. Importance of grayscale conformity loss and each of its sub-terms.

In our case, we require the latent sparse representation to be visualizable as a grayscale image that conforms to the color input.

The detailed architectures of our proposed encoding and decoding networks are illustrated in Fig. 2, with key parameters annotated below each block. The encoding network contains two down-convolution blocks, two up-convolution blocks, eight residual blocks [He et al. 2016], and two flat convolution layers. Here, the down/up convolution structure is used to increase the receptive field for feature extraction. The decoding network contains eight residual blocks and one flat convolution layer. We only adopt down/up-convolution structure in the encoding network because richer neighborhood context is needed for encoding color information in grayscale, while decoding the grayscale back to color is relatively local. Along with using the down/up-convolution structure, we also adopt the skipping strategy [Ronneberger et al. 2015] in the encoding network to suppress the blurring artifacts during downscaling/upscaling. Fig. 3 presents a comparison between not using and using down/up-convolution structure. As presented in Fig. 3(b), significant artifacts can be observed on the wall without using down/up-convolution structure.

5 LOSS FUNCTION

As in Eq. 3, our loss function consists of three terms: invertibility loss $\mathcal{L}_V(E, D)$, grayscale conformity loss $\mathcal{L}_C(E)$ and quantization loss $\mathcal{L}_Q(E)$.

5.1 Invertibility Loss

Given an input color image I and the restored color output $R = D(E(I))$, the invertibility loss $\mathcal{L}_V(E, D)$ ensures that I and R are as similar as possible. We simply utilize a per-pixel mean square error (MSE) to regularize this similarity:

$$\mathcal{L}_V(E, D) = \mathbb{E}_{I\in\mathcal{I}}\{||R - I||_2\} \tag{4}$$

Here, $||\cdot||_2$ denotes the L_2 norm (MSE), and $\mathbb{E}$ denotes the average operator over all images in the training dataset $\mathcal{I}$. As R is generated via both the encoder E and the decoder D, this loss effectively imposes constraints over the parameters of both E and D.

5.2 Grayscale Conformity Loss

Given the color input I and the converted grayscale $G = E(I)$, the conformity loss $\mathcal{L}_C(E)$ ensures that the grayscale image G visually conforms to the original color input I, and matches the general expectation on how a grayscale image looks like. Here, we emphasize again that there exists no absolute standard of grayscale conversion. Simple grayscale representation adopts the L channel in the CIE *Lab* color space or the Y channel in the YIQ/YUV color space as grayscale, which does not emphasize preserving regional contrast. Some decolorization methods [Liu et al. 2015; Lu et al. 2012] adopt more sophisticated computation to preserve the contrast. In our case, our primary goal is to encode the color information in the form of unobvious pattern. We need a relatively relaxed form as our grayscale conformity loss, instead of solely constraining the MSE. In particular, our grayscale conformity loss consists of three sub-terms: lightness loss, contrast loss, and local structure loss.

5.2.1 Lightness Loss. The lightness loss ensures that the generated grayscale image G visually conforms to the luminance of I. That is, bright regions remain bright, and dark regions remain dark. However, we need to leave a room for color encoding. This is achieved by permitting a grayness difference θ in the formulation. In particular, the lightness preservation loss $\ell_l(E)$ is regarded as 0 if the difference is smaller than a threshold θ:

$$\ell_l(E) = \mathbb{E}_{I\in\mathcal{I}}\{||\max\{|G - L(I)| - M_\theta, M_0\}||_1\} \tag{5}$$

Here, $||\cdot||_1$ denotes the L_1 norm. $|\cdot|$ is an element-wise absolute value operator. $L(I)$ denotes the luminance channel of I. $\max\{\cdot,\cdot\}$ is an element-wise maximum operator between two matrices with the same size. M_θ/M_0 is a matrix with the same size as G where every element in M_θ/M_0 is equal to $\theta/0$. With pixel values in I and G defined in $[0, 255]$, we empirically set $\theta = 70$ in all our experiments. Note that this requirement is extremely loose to allow a larger searching space.

5.2.2 Contrast Loss. The contrast loss aims at preserving the global contrast of input color I in the resultant grayscale G. We find that high-level features defined in the pretrained VGG-19 network [Simonyan and Zisserman 2014] work extremely well in representing the global contrast. Therefore, we define our contrast loss $\ell_c(E)$ as the similarity between the corresponding VGG layers of I and G as

$$\ell_c(E) = \mathbb{E}_{I\in\mathcal{I}}\{||\mathrm{VGG}_k(G) - \mathrm{VGG}_k(I_c)||_1\} \tag{6}$$

Here, $\mathrm{VGG}_k(\cdot)$ denotes the k-th VGG layer extracted from an image. In particular, we use the layer 'conv4_4' of G and I to enforce the similarity of global contrast. c is the color channel index of I, taking values in $\{1, 2, 3\}$.

5.2.3 Local Structure Loss. The last sub-loss aims at preserving the local structure of the color input I in the grayscale G. That is, if a pixel is locally smooth in the input, it should remain smooth in the generated grayscale image. This can prevent the color-encoding pattern in the grayscale image being apparent. To measure the structure similarity, we adopt the local variation which is simple but effective. So the local structure loss $\ell_s(E)$ can be formulated as

$$\ell_s(E) = \mathbb{E}_{I \in \mathcal{I}}\{||\mathrm{Var}(G) - \mathrm{Var}(I))||_1\} \tag{7}$$

where $\mathrm{Var}(\cdot)$ is a function that calculates the mean of local variation of an image.

5.2.4 Combined Loss. The overall grayscale conformity loss is a weighted sum of the above three sub-terms:

$$\mathcal{L}_C(E) = \ell_l(E) + \alpha\ell_c(E) + \beta\ell_s(E) \tag{8}$$

where α and β are weighting factors. We empirically set $\alpha = 1^{-7}$ to balance the magnitude difference between the convolutional feature space and the image pixel space. The weight $\beta = 0.5$ is used to balance between the local structure loss and other losses. In fact, we found that the training result is not sensitive to the hyper parameters. Therefore, fixed values are adopted throughout all experiments.

Fig. 4 shows the importance of the grayscale conformity loss and each sub-term. Without the grayscale conformity loss, the resultant grayscale in (b) does not look like a grayscale image of the color input. If we drop the lightness term, the contrast may be flipped as shown on the petals in (c). If the global contrast term is dropped, the overall contrast of the color input cannot be preserved in the grayscale, leading to a "gray" result as shown in (d). If we drop the local structure term, the model cannot effectively suppress the color-encoding pattern as exhibited in (e) (compared to the corresponding blow-up in (f)). Only when all terms are included, a visually appealing grayscale, without obvious color encoding pattern, is obtained in (f).

5.3 Quantization Loss

The precision of images (grayscale or color) we discussed so far is 32-bit floating-point. In real practices, each pixel in a grayscale image is stored in an 8-bit unsigned integer precision. This quantization error may lead to artifacts when the quantized grayscale is restored to its color version, as demonstrated in Fig. 5(a). Therefore, we propose to penalize all non-integer pixel values via the following quantization loss:

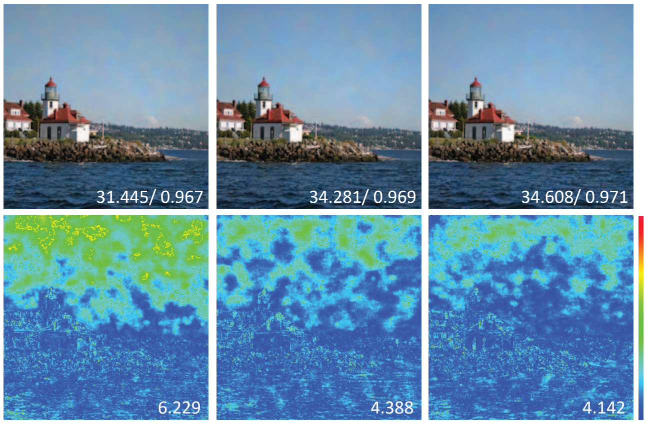
$$\mathcal{L}_Q(E) = \langle || \min_{d=0}^{255}\{|G - M_d|\}||_1 \rangle_{\mathcal{I}} \tag{9}$$

where $\min\{\cdot\}$ is an element-wise minimum operator among a set of matrices with the same size. M_d is a matrix with the same size as G where every element in M_d is equal to d. By incorporating the quantization loss, we can suppress the artifact introduced by quantization and achieve significantly better results (Fig. 5(b)).

6 TRAINING

6.1 Training Data

Since the groundtruth of the output is exactly the same as the input, our system can actually take any color image I and use $\langle I, I\rangle$ as the training pair for supervised training. In our experiment, we use the VOC2012 (Visual Object Classes Challenge 2012) dataset [Everingham et al. 2012] for training and testing. There are 17,125 color images in the dataset in total. Among them, 13,758 are used for training while the rest images are used for testing. Even though our data size is not very large, experiments show that we can achieve extremely high-quality restoration results (average SSIM of 0.9681 and average PSNR of 36.02 dB). The performance of our system may be further improved with a larger dataset, but very slightly. All input images are cropped and resized to 256 × 256 resolution during training, but images of arbitrary resolutions can be processed during the testing.



(a) Without $\mathcal{L}_Q$ (b) 1-stage scheme with $\mathcal{L}_Q$ (c) 2-stage scheme with $\mathcal{L}_Q$

Fig. 5. Importance of quantization loss. The top row shows the restored color images, and the bottom row shows the pixel-wise color difference map. The values of the difference map are in [0, 32]. (a) Without quantization loss, apparent color change can be observed in the sky. (b) & (c) With quantization loss, artifacts can be successfully suppressed in both 1-stage and 2-stage training, while 2-stage training leads to slightly better result.

Table 1. Hyper-parameters of our two-stage training scheme.

Stage	ω_1	ω_2	Epochs
I	1.0	0.0	90
II	0.5	10.0	30

6.2 Two-stage Training

Our model is trained end-to-end with the loss function defined in Eq. 3. We may train the network from scratch (one-stage scheme), and the training loss against actual running time is plotted as the green curve in Fig. 6. However, the quantization loss is a piecewise function which is both difficult to train and occupying large memory space. Hence, instead of training from scratch, we also propose a two-stage training scheme to accelerate the training and reduce memory consumption. In the first stage, we only consider the invertibility loss and grayscale conformity loss. In the second stage, all three losses are accounted. To achieve this, we set ω_2 in Eq. 3 to 0 to omit the quantization loss in the first 90 epochs (the first stage), and set it to a relatively larger value to mainly focus on the quantization loss in the subsequent 30 epochs (the second stage). Weights α and β in Eq. 8 remain unchanged in both stages. The detailed parameter values are tabulated in Table 1. The training loss of the two-stage scheme against actual running time is plotted as the red curve in Fig. 6. Each dot indicates an epoch. The two-stage training converges much faster (in actual time) than that of the one-stage training. Even though the number of epochs are the same, the two-stage training consumes less amount of total time. The restored color image trained with the two-stage training is also slightly better than that of the one-stage training (Fig. 9(b) & (c) and Table 2).

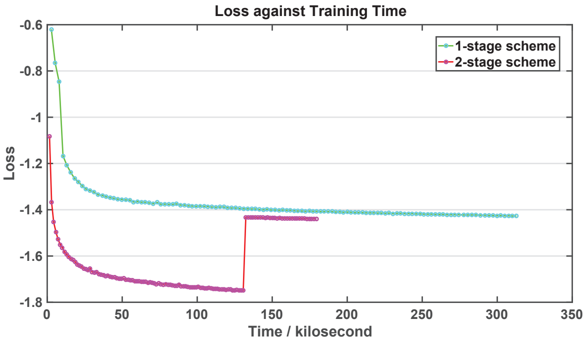


Fig. 6. Loss curves of our 2-stage training and training from scratch (1-stage training). We plot the whole training process from epoch 1 to 120 for both loss curves (the y-axis is on a logarithmic scale). We can observe that our two-stage training is much faster than training from scratch.

During training, the learning rate is initialized to 0.0002, and then linearly decreased to 0.000002 through the 120 epochs. Our model is optimized by the ADAM solver [Kingma and Ba 2014].

7 RESULTS AND DISCUSSION

7.1 Qualitative Evaluation

We evaluated our method on images of various genre and content, convincing results are obtained in all cases. Fig. 14 showcases the invertible grayscale and restored color images of multiple categories, from portrait closeup to faraway scenery, and from rural to metropolitan. For each example, we show the original input (the first column), the invertible grayscale with 8-bit quantization (the second column), the color image restored from the quantized grayscale, and the color-coded difference map (the fourth column). The PSNR and SSIM between the restored and original color images are annotated in the restored color images. The mean absolute error (MAE) between the restored and original color images are annotated in the difference maps. Readers are encouraged to zoom in the images to inspect the fine details. The generated grayscale conforms to the color input and resembles what we normally expect for a grayscale image. Unless substantially zooming into the grayscale image, it is hard to recognize the unobvious color-encoding pattern. For busy regions, it is even harder to recognize any pattern. The restored color images are almost the same with the original color input. More examples can be found in the supplementary materials.

7.1.1 Color Restoration vs. Colorization. We first compare our results to that from the state-of-the-art CNN-based colorization method [Zhang et al. 2016] in Fig. 7. However, we have to emphasize that this comparison may not be very appropriate. The input grayscale image fed to Zhang method is the L channel of the color input, while the grayscale image fed to our method is the invertible grayscale. Since the colorization method heavily relies on the correlation of color and natural textures, it generally fails when the image lacks of textures as in Fig. 7(b). In sharp comparison, our method successfully restores the original colors using the encoded color information in the invertible grayscale (Fig. 7(c)). We further compare our method to the state-of-the-art CNN-based *interactive* colorization method [Zhang et al. 2017] which colorizes grayscale images based on user color hints. For each image, we generate results from their method with 500 randomly sampled color-hint points, as suggested in their paper. Fig. 8 compares their results and ours. Even with densely sampled color hints, their results may still have noticeable differences from the groundtruth.

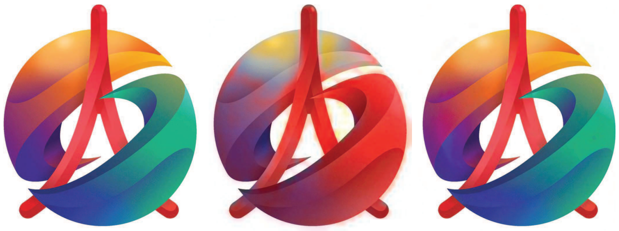

(a) Groundtruth (b) [Zhang et al. 2016] (c) Ours

Fig. 7. Color restoration vs. colorization. (a) Input. (b) Colorized result generated by colorizing the luminance channel of (a) with a CNN-based colorization method [Zhang et al. 2016]. (c) Color image restored from our invertible grayscale image.

Table 2. Quantitative evaluation on invertibility in terms of MAE, PSNR, and SSIM. Lower MAE and higher PSNR/SSIM indicate better invertibility.

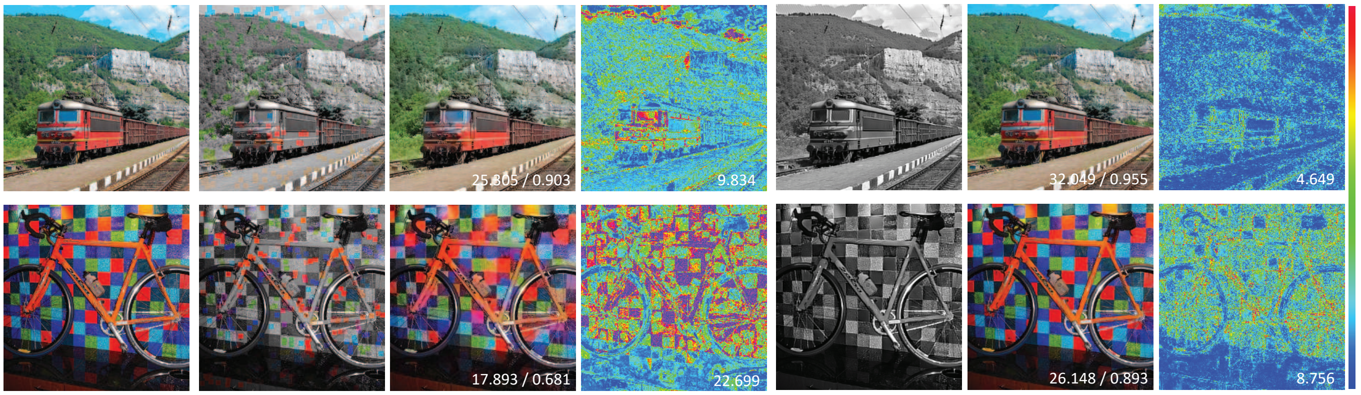
Mode	MAE		PSNR		SSIM	
	Mean	Stddev	Mean	Stddev	Mean	Stddev
Zhang et al. 2017	6.682	1.584	29.72	2.183	0.9375	0.03970
1-stage	3.345	1.089	35.44	2.499	0.9641	0.02271
2-stage	**3.083**	1.055	**36.02**	2.666	**0.9681**	0.02007

7.1.2 Color-Encoding Pattern. The side effect of encoding color information in the invertible grayscale is the introduction of unobvious pattern (Fig. 1 & 9). It is only observable when the image is significantly zoomed in. In general, the pattern is very complex to understand. It is very slight in bright regions and relatively apparent in darker regions. It is also more obvious in smooth regions than in busy regions, probably due to the fact that human visual system is less sensitive to the change of busy visual content. We also find that the scale of the generated pattern is independent of the input resolution. Fig. 9 shows an example where the input (Fig. 9(a)) is rescaled to two resolutions, 400×300 (Fig. 9(b)) and 1,600×1,200 (Fig. 9(c)). The high-resolution one better hides the color-encoding pattern as further zooming in is needed to recognize the pattern.

If our model is fed with a grayscale input, no color-encoding pattern is introduced in the invertible grayscale, as there is no color information to encode. The bottom row of Fig. 10 compares the effect of color and grayscale input to our system. Comparing to the color input case in which pattern is introduced (Fig. 10(c)), no pattern is observable in Fig. 10(f). The invertible grayscale is almost the same as the input grayscale. In addition, our invertible grayscale image (Fig. 10(b)) can better preserve the original contrast, compared to the L image (Fig. 10(e)), due to our contrast loss. But our method is inferior in terms of color-contrast preservation ability when compared to existing contrast-preserving decolorization methods [Gooch et al. 2005; Lu et al. 2012]. We have to emphasize, the primary goal of our method is invertiblity, even though we can preserve the global contrast to certain extent.

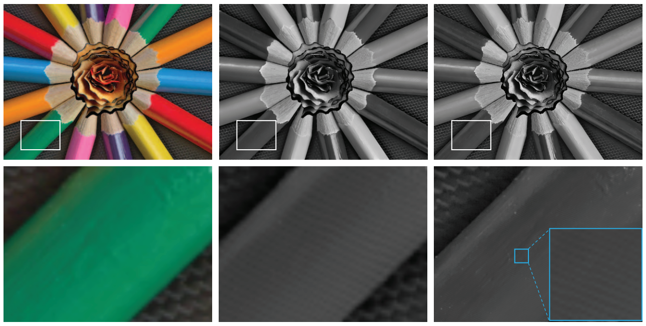
7.2 Quantitative Evaluation

We further evaluate our system quantitatively on the testing dataset composed of 3,367 images. To measure the similarity between the



(a) Ground truth (b) Grayscale with hints (c) Colorized result ([Zhang et al. 2017]) (d) Difference map of (c) (e) Invertible grayscale (f) Restored result (Ours) (g) Difference map of (f)

Fig. 8. Comparison to [Zhang et al. 2017]. (a) Input. (b) Grayscale image of (a) overlaid with 500 randomly sampled color-hint points. (c) Colorized results generated by [Zhang et al. 2017] using the color-hint points as shown in (b). (d) Pixel-wise difference map between (c) and (a). (e) Our generated grayscale images of (a). (f) Color images restored from the invertible grayscale images as shown in (e) using our method. (g) Pixel-wise difference map between (f) and (a). The PSNR/SSIM with respective to the groundtruth are labelled in the colorization results. The difference maps (d)&(g) are plotted in the scale of [0, 32]. The corresponding MAE is labelled in each difference map.

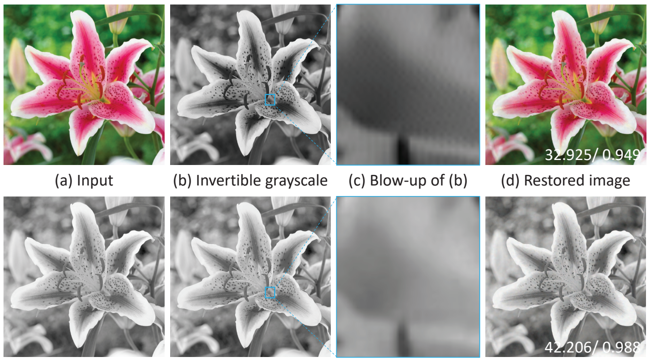


(a) Input (b) Resolution: 300 x 400 (c) Resolution: 1200 x 1600

Fig. 9. The scale of the generated pattern is independent of the resolution of the input. The high-resolution image can better hide the color-encoding pattern as further zooming in is needed in order to recognize the pattern.

restored and original color images, we adopt three metrics: mean absolute error (MAE), PSNR, and SSIM [Wang et al. 2004], which are the average values of corresponding metrics computed in RGB channels. Similarly, we also compare with [Zhang et al. 2017] where 500 randomly sampled color-hint points are used for each image. The statistics is tabulated in Table 2. For the MAE metric, the value is in the range of [0,255], lower values indicate better similarity, while 0 indicates exactly the same. For PSNR and SSIM metrics, higher values indicate better similarity, where ∞ and 1 indicate the best, respectively. Compared to [Zhang et al. 2017], our method achieves a much lower MAE and much higher scores for both PSNR and SSIM metrics, which means our restored color images are significantly closer to the groundtruth than the colorized ones.

There is no standard way to measure the conformity of a grayscale image to the original color image. We measure the structure similarity between the generated grayscale and the L channel of the color input using the structure term defined in SSIM [Wang et al. 2004]. The rationale we take the L channel as groundtruth is because it mostly captures the structural information of the color input. In the range of [-1, 1], the average and standard deviation are 0.9799 and 0.008830, respectively. This shows that our invertible grayscale well preserves the structure information of the input image.



(a) Input (b) Invertible grayscale (c) Blow-up of (b) (d) Restored image

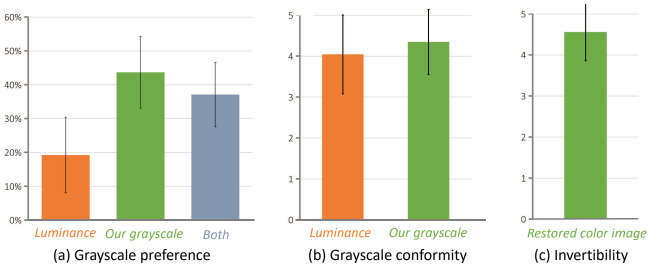
(e) Input (f) Invertible grayscale (g) Blow-up of (f) (h) Restored image

Fig. 10. Comparison between color and grayscale input. Our system generally introduces unobvious pattern in the generated grayscale as the side effect of encoding color information, as shown in (c). However, for grayscale input, no color-encoding pattern is introduced since there is no color information to encode, as shown in (g).

7.3 User Study

We further conduct a user study to evaluate our system in terms of human-perceived invertibility and grayscale conformity. We invited 25 participants, including 15 males and 10 females, aging from 18 to 33. We randomly select 14 example images of different categories for the user study. The detailed questionnaire and example images can be found in the supplementary material.

7.3.1 Invertibility. To evaluate how well our invertible grayscale preserves the original colors, the input and restored color image are shown side-by-side to the participants. Then we ask each participant to rate the similarity between the two images in the scale of 0 to 5, with 0 being completely different and 5 being exactly the same.



(a) Grayscale preference (b) Grayscale conformity (c) Invertibility

Fig. 11. Statistic result of the user study.

Table 3. Timing statistics (in seconds).

Image Size	CPU only		With GPU	
	Encoding	Decoding	Encoding	Decoding
256 × 256	0.939	0.825	0.034	0.029
512 × 512	3.717	3.284	0.113	0.086
1024 × 1024	15.117	13.133	0.384	0.321

The result is shown in Fig. 11(c). Most participants think that our restored color image is very similar to the input image.

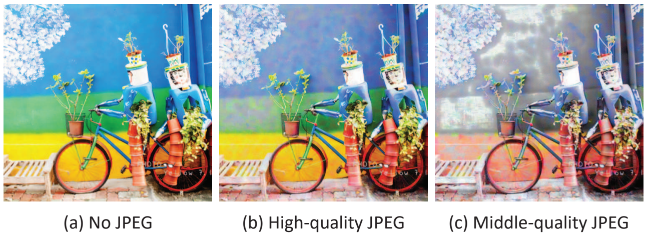
7.3.2 Grayscale Conformity. To evaluate how well our invertible grayscale conforms to the color input, we side-by-side show the color input, L channel of the input, and our invertible grayscale image to the user. Note that the positions of the two grayscale images, luminance image and our invertible grayscale, are randomly interchanged to avoid bias. We then ask the participants to choose the grayscale that better represents the original color input. We also allow them to choose both if they think both grayscale images are similar. Fig. 11(a) shows the results. In general, our invertible grayscale is more preferred since it better preserves the image contrast. We further ask the participants to give a rating on the similarity of each grayscale comparing to the input from 0 to 5, with 0 being completely different and 5 being exactly the same. Fig. 11(b) shows the results. Both grayscale images receive high ratings while ours is slightly better than the luminance image.

7.4 Timing Statistics

We implement our model using TensorFlow [Abadi et al. 2016] with Python. All experiments were performed on a PC with Intel Xeon E5-1630 v4 3.70GHz CPU and GeForce GTX 980 Ti GPU. We evaluate the running time of both the encoding and decoding networks for images of different resolutions with and without using GPU. For each image resolution, we run the experiment for 100 times and take their average. The timing statistics is tabulated in Table 3. In general, the encoding takes more time than the decoding, since the encoding network consists of more convolution layers and involves more computation. With GPU, our method achieves real-time performance for 512 × 512-resolution images.

7.5 Limitations

Since our invertible grayscale system encodes the original color information as unobvious patterns in the grayscale, the quality of the restored color image highly depends on the accuracy of these patterns. Therefore, the generated invertible grayscale image is not resistant to general image manipulations, such as image rotating, image resizing, and JPEG compression, since the image manipulation operators may ruin the unobvious color-encoding patterns. Fig. 12 shows the restored color images when the invertible grayscale image is stored losslessly (Fig. 12(a)), with high-quality JPEG compression (Fig. 12(b)), and with mid-quality JPEG compression (Fig. 12(c)). Lowering the compression quality of the grayscale image further damages the color-encoding patterns, and hence lowers the quality of the restored color image. To strengthen the robustness to certain image manipulations, a potential solution might be incorporating the manipulations in the model during training, which is also a potential direction of our future work.



(a) No JPEG (b) High-quality JPEG (c) Middle-quality JPEG

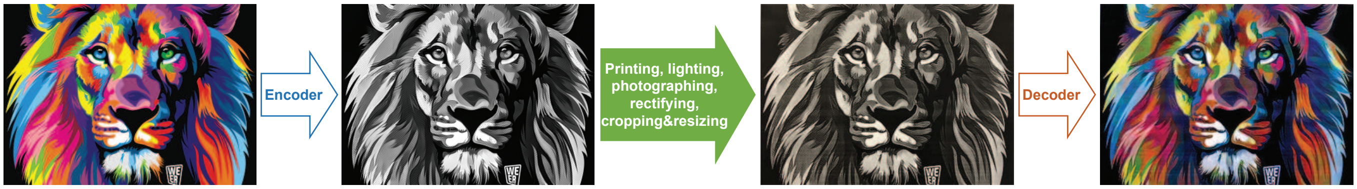
Fig. 12. Limitation w.r.t. JPEG compression. Note that we present the restored color image here while JPEG compression is applied on the grayscale image. (a) Without JPEG compression. (b) With high-quality JPEG compression. (c) With middle-quality JPEG compression.

Nevertheless, our current system does tolerate certain degree image contamination, as demonstrated in our supplementary video. In this video, we first generated the invertible grayscale images using our encoding network and printed them on regular A4 paper. Then we took photographs of them with a handheld camera equipped on a PDA. The acquired grayscale images may contain noise, blurriness, lighting variation, and even perspective distortion. Finally, we restored the color images from these noise-contaminated grayscale images using our decoding network. As demonstrated in the supplementary video as well as in Fig. 13, convincing results can still be obtained, although the restored color may slightly deviate from the groundtruth.

8 CONCLUSION

In this paper, we proposed an innovative method to convert a color image to grayscale, that can be later inverted back to its color version. The key idea is to encode the color information into the synthesized grayscale. We proposed to learn and embed the color-encoding scheme in a convolutional neural network. Given arbitrary color input, the learned network can generate visually pleasant grayscale images and obtain the color-restored images that are hardly differentiable from the original color input. All quantitative experimental results and user statistics support the validity of our method.

The proposed method is designed for still pictures. Although one can apply it to videos in a per-frame generation manner, the current model may not be able to guarantee the temporal consistency. Further study on such temporal consistency is needed. Another direction is on its noise resistance. Currently, it is not resistant to error introduced by lossy image compression such as JPEG. One potential solution is to model the compression error into the neural networks, so that the trained model can learn to encode color information in a more noise-resistant fashion.


Encoder
Printing, lighting, photographing, rectifying, cropping&resizing
Decoder

(a) Input (b) Invertible grayscale (c) Reacquired grayscale (d) Restored image

Fig. 13. Tolerance to contamination. We printed our generated grayscale (b) on paper, took a photograph of it (c), and finally restored the color image (d). From (b) to (c), there can be a series of contamination-inducing operations, including printing, lighting, photographing, image rectification, cropping and resizing.

While our current method is tailored for the invertible color-to-gray problem, the proposed general framework can be applied to many other applications requiring information encoding in images. One immediate extension is to convert color/grayscale image to invertible halftone which is more suitable for printing purposes.

ACKNOWLEDGMENTS

This project is supported by Shenzhen Science and Technology Program (No.JCYJ20160429190300857) and Shenzhen Key Laboratory (No.ZDSYS201605101739178), and the Research Grants Council of the Hong Kong Special Administrative Region, under RGC General Research Fund (Project No. CUHK14201017).

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A system for large-scale machine learning.. In *OSDI*, Vol. 16. 265–283.

Xiaobo An and Fabio Pellacini. 2008. AppProp: all-pairs appearance-space edit propagation. *ACM Transactions on Graphics (TOG)* 27, 3 (2008), 40.

Mohammad Haris Baig and Lorenzo Torresani. 2017. Multiple hypothesis colorization and its application to image compression. *Computer Vision and Image Understanding (CVIU* 164 (2017), 111–123.

Raja Bala and Reiner Eschbach. 2004. Spatial color-to-grayscale transform preserving chrominance edge information. In *Color and Imaging Conference*.

Miguel A Carreira-Perpinán and Ramin Raziperchikolaei. 2015. Hashing with binary autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based photo recoloring. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 139.

Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. 2008. Automatic image colorization via multimodal predictions. In *European Conference on Computer Vision (ECCV)*.

Zezhou Cheng, Qingxiong Yang, and Bin Sheng. 2015. Deep colorization. In *IEEE International Conference on Computer Vision (ICCV)*.

Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin. 2011. Semantic colorization with internet images. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 156.

Aditya Deshpande, Jason Rock, and David Forsyth. 2015. Learning large-scale automatic image colorization. In *IEEE International Conference on Computer Vision (ICCV)*.

Sovann En, Bruno Crémilleux, and Frédéric Jurie. 2017. Unsupervised deep hashing with stacked convolutional autoencoders. (2017).

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2012. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. (2012).

Amy A Gooch, Sven C Olsen, Jack Tumblin, and Bruce Gooch. 2005. Color2gray: salience-preserving color removal. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 634–639.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference Computer Vision and Pattern Recognition (CVPR)*.

Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. 2005. An adaptive edge detection based colorization algorithm and its applications. In *ACM International Conference on Multimedia (MM)*.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 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 (TOG)* 35, 4 (2016), 110.

Revital Irony, Daniel Cohen-Or, and Dani Lischinski. 2005. Colorization by Example. In *Rendering Techniques*.

Yongjin Kim, Cheolhun Jang, Julien Demouth, and Seungyong Lee. 2009. Robust color-to-gray via nonlinear global mapping. *ACM Transactions on Graphics (TOG)* 28, 5 (2009), 161.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1511.06349* (2014).

Jung Gap Kuk, Jae Hyun Ahn, and Nam Ik Cho. 2010. A color to grayscale conversion considering local and global contrast. In *Asian Conference on Computer Vision (ACCV)*.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Learning representations for automatic colorization. In *European Conference on Computer Vision (ECCV)*.

Anat Levin, Dani Lischinski, and Yair Weiss. 2004. Colorization using optimization. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 689–694.

Qiegen Liu, Peter X Liu, Weisi Xie, Yuhao Wang, and Dong Liang. 2015. GcsDecolor: gradient correlation similarity for efficient contrast preserving decolorization. *IEEE Transactions on Image Processing (TIP)* 24, 9 (2015), 2889–2904.

Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. 2008. Intrinsic colorization. *ACM Transactions on Graphics (TOG)* 27, 5 (2008), 152.

Cewu Lu, Li Xu, and Jiaya Jia. 2012. Contrast preserving decolorization. In *IEEE International Conference on Computational Photography (ICCP)*.

Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. 2007. Natural image colorization. In *Eurographics Conference on Rendering Techniques*.

Laszlo Neumann, M Čadík, and Antal Nemcsics. 2007. An efficient perception-based adaptive color to gray transformation. In *Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*.

Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. 2006. Manga colorization. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 1214–1220.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*.

Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2017. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Mingli Song, Dacheng Tao, Chun Chen, Xuelong Li, and Chang Wen Chen. 2010. Color to gray: Visual cue preservation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 9 (2010), 1537–1552.

Baoyuan Wang, Yizhou Yu, Tien-Tsin Wong, Chun Chen, and Ying-Qing Xu. 2010. Data-driven image color theme enhancement. *ACM Transactions on Graphics (TOG)* 29, 6 (2010), 146.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing (TIP)* 13, 4 (2004), 600–612.

Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. 2002. Transferring color to greyscale images. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 277–280.

Kun Xu, Yong Li, Tao Ju, Shi-Min Hu, and Tian-Qiang Liu. 2009. Efficient affinity-based edit propagation using kd tree. *ACM Transactions on Graphics (TOG)* 28, 5 (2009), 118.

Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. 2017. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 119.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*.

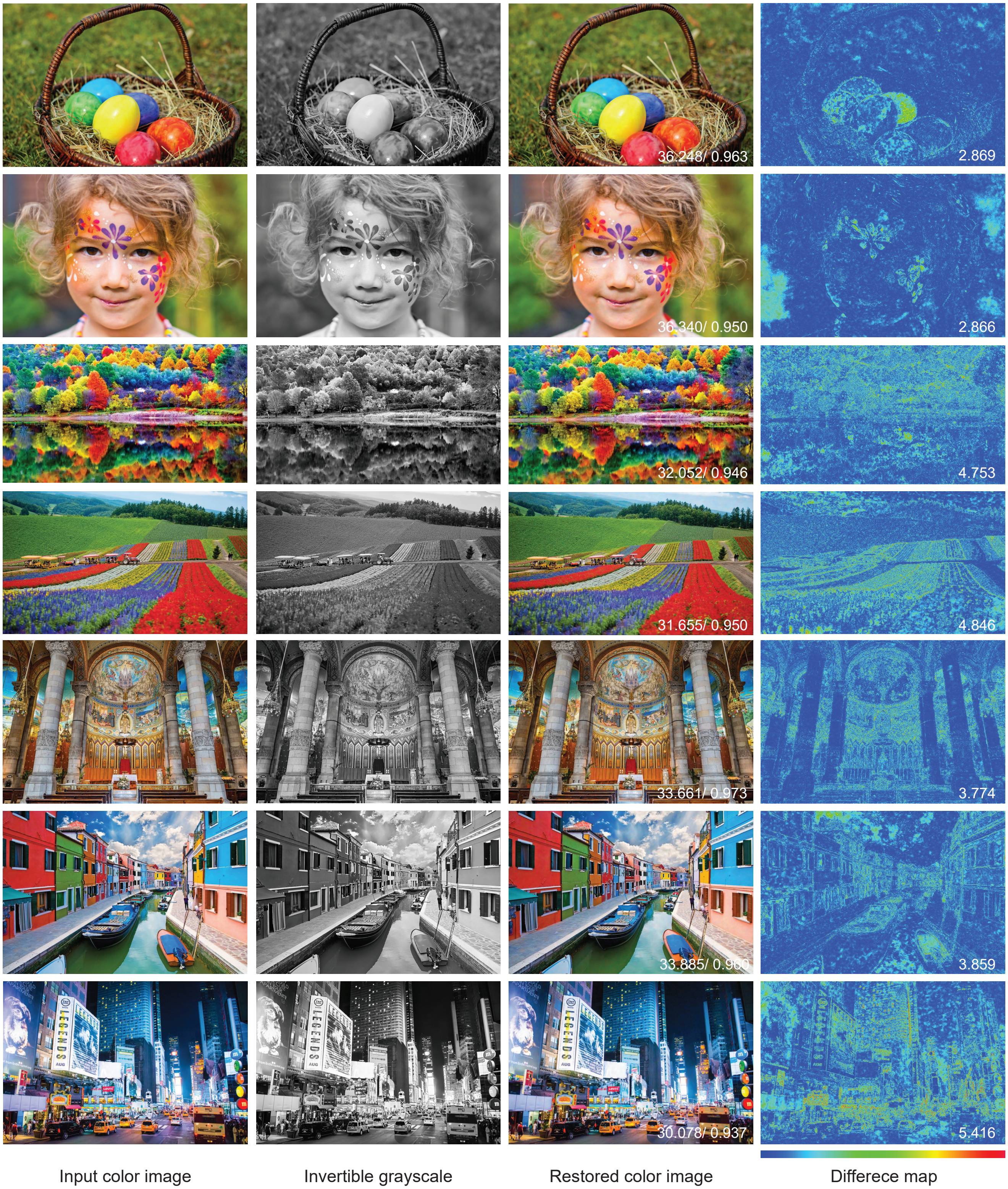


Fig. 14. Result gallery. The PSNR/SSIM with respective to the groundtruth are annotated in the restored color images. The difference maps between the restored color images and the groundtruth are illustrated in the range [0, 32]. The MAE with respective to the groundtruth are annotated in the difference maps.