Neural Image Recolorization for Creative Domains (Supplementary Material)

Boyi Li
Cornell University

Serge Belongie
University of Copenhagen

Ser-nam Lim
Meta AI

Abe Davis
Cornell University

Appendix

A. Loss Function

The loss for our exemplar-based approach has 3 parts: GAN adversarial loss [1] L_{adv}, latent regression loss L_{idt}, and cycle consistency loss L_{cycle}. During training, we have inputs I_A and reference I_B, extracted color latent vectors C_A and C_B, corresponding normalization constants β_A, γ_A and β_B, γ_B, as well as output I'_A:

$$I'_A = ColorHouse(I_A, C_B). \tag{1}$$

Adversarial loss is the standard in GANs for matching the distribution of the translated image to the target domain:

$$L_{adv} = \mathbb{E}[\log D(I_A)] + \mathbb{E}[1 - \log D(I'_A)))] \tag{2}$$

Here we have the discriminators D for distinguishing between real and generated images for domain A.

We use L_{idt} to ensure the recolorization autoencoder is able to reconstruct the input when given the original input as exemplar:

$$L_{idt} = \mathbb{E}[||I_{AA} - I_A||_1], \tag{3}$$

where

$$I_{AA} = ColorHouse(I_A, C_A).$$

L_{cycle} is based on Cycle-Consistency loss [8] that optimizes the training for this under-constrained problem and regularizes the translated image to preserve semantic structure of the input image:

$$L_{cycle} = \mathbb{E}[||I'_{cycleA} - I_A||_1], \tag{4}$$

where

$$I'_{cycleA} = ColorHouse(I'_A, C_A).$$

Our final objective function is:

$$L_{exemplar-based} = \lambda_{adv} L_{adv} + \lambda_{idt} L_{idt} + \lambda_{cycle} L_{cycle}, \tag{5}$$

where λ_{adv}, λ_{idt}, λ_{cycle} are weights assigned for each loss, respectively.

Color palette-based ColorHouse uses the same loss function as the exemplar-based method. Different from previous methods, to benefit from the superior performance of exemplar-based approaches, we feed the palette (including illumination) into the network to obtain the color latent vectors L_p and push it towards the latent code L_e from the fixed exemplar-based branch.

$$L_l = \mathbb{E}[||L_p - L_e||_1], \tag{6}$$

Additionally, we utilize palette L_1 loss $L_{palette-based}$ to push the color palettes extracted from the generated outputs to match that of the input:

$$L_{palette} = \mathbb{E}[||PalettePredictor(I'A) - PalettePredictor(I_A)||_1], \tag{7}$$

We utilize illumination L1 loss L_{illum} to push the predicted illumination to be the same with the one of the input:

$$L_{illum} = \mathbb{E}[||IllumPredictor(I'A) - IllumPredictor(I_A)||_1], \tag{8}$$

Therefore, the final objective function is:

$$\begin{aligned} L_{palette-based} = \lambda_{adv} L_{adv} + \lambda_{idt} L_{idt} + \lambda_l L_l + \\ \lambda_{cycle} L_{cycle} + \lambda_{palette} L_{palette} + \lambda_{illum} L_{illum}, \end{aligned} \tag{9}$$

where λ_{adv}, λ_{idt}, λ_{cycle}, $\lambda_{palette}$ and λ_{illum} are the weights assigned for each loss.

Layer Specs: Our generator consists of one 5×5 stride-1, one 3×3 stride-2, and one 3×3 stride-1 convolutional layer. We extract the latent vector after the second and third layer. Our recolorization backbone is composed of an encoder and a decoder. We insert the first color latent vector into the encoder and the second to the decoder. The encoder consists of one 7×7 stride-1, one 5×5 stride-1, one 3×3 stride-2 and one 3×3 stride-1 convolutional layer. The decoder consists of one 3×3 stride-1 as well as one 3×3 stride-2 deconvolutional layer, following with one 5×5 stride-1 and one 7×7 stride-2 convolutional layer.

ColorHouse palette predictor

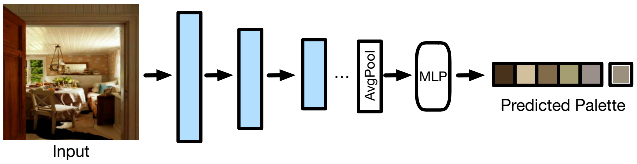


Figure 1: ColorHouse palette predictor.

B. Additional Experiments

B.1. Experiment Implementation

We use collected interior design images from the Internet (around 3,000) and for artwork images we used Behance Artistic Media dataset (BAM) [5], a large-scale dataset of contemporary artwork. During training, we use Adam with 0.0005 learning rate, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The weights are initialized using Gaussian random variables. Empirically, for training, we find the model starts to converge after 10 epochs and become stable after 100 epochs using a single GPU. For testing, exemplar-based ColorHouse runs on a 256×256 image in about 0.004 seconds, and color palette-based ColorHouse runs in about 0.008 seconds, using a single Geforce GTX 1080 Ti GPU without acceleration. This performance is very fast, suggesting that ColorHouse could eventually be used in real time interactive applications. Also, unlike many previous methods (e.g, those trained with ImageNet), ColorHouse can produce natural-looking outputs even when trained with limited data. This potentially results from our use of several different semantically meaningful loss functions to optimize the training process. We describe our implementation in the supplemental material, and we will post extensive additional images and metadata to support its use upon publication.

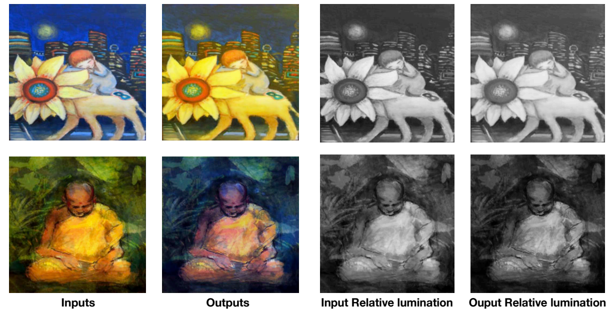
Exemplar-based luminance adaptation. Colorization often assumes that the luminance of the input matches the luminance of output [2, 6, 7]. However, recolorization attempts to map between colors that may vary in this respect. Therefore, ColorHouse operates on RGB space instead of Lab color space [3]. In Fig. 2, we show an example of color swapping by exemplar-based ColorHouse and the relative luminance before and after applying ColorHouse. Relative luminance (Y) is a widely used metric that relates to the luminous flux density in a particular direction and can be approximated from linear RGB components [4]:

$$Y = 0.2126R + 0.7152G + 0.0722B.$$

We can observe that ColorHouse modifies luminance in ways that most colorization or recolorization methods would not allow, demonstrating the expressive power of our approach.

B.2. Interior Design

We display the exemplar-based results for interior design photos in Fig. 3. In detail, Fig. 3(a) shows the results of same inputs with various references, Fig. 3(b) shows the results for fixed references across various inputs. We observe that ColorHouse works very well for diverse inputs and references with promising generated images. For example, in the second row of Fig. 3(a), ColorHouse successfully learns the diverse color styles from different reference images and recolorizes the library photos to the corresponding color.



(a) Exemplar-based Recolorization. (b) Relative luminance.

Figure 2: Exemplar-based Color Swapping and its relative luminance outputs.

(a) Same inputs, various references.

(b) Same references, various inputs.

Figure 3: Exemplar-based ColorHouse results on interior design photos.

We display the palette-based results in Fig. 4.[1] To make the task harder, we select diverse color palettes. For example, the first palette consists of 5 totally different colors, the second consists of pink or red. Then we feed different inputs and color palettes to the model, and we observe that ColorHouse can generate desirable outputs based on different color palettes.

B.3. Recolorized Art

Beyond the domain of interior design, ColorHouse is also well suited to artwork. To illustrate this, we conduct experiments on Behance Artistic Media dataset (BAM) [5].

[1] The following experiments follow the same setting.

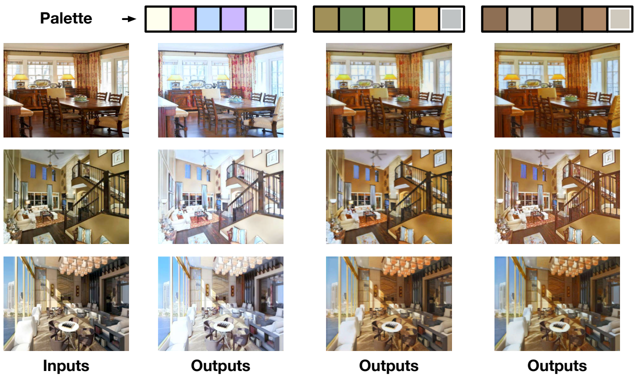


Figure 4: Color palette-based ColorHouse results on interior design photos.

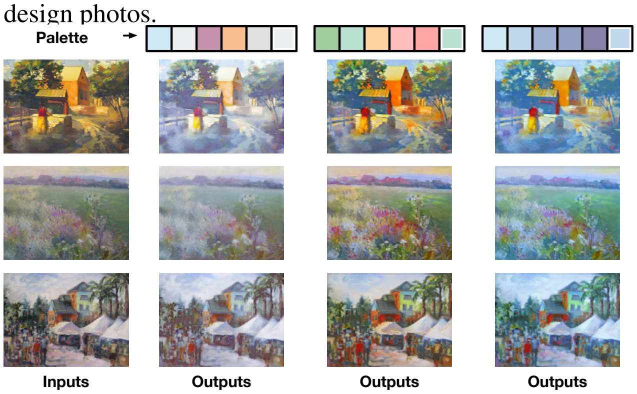


Figure 5: Color palette-based ColorHouse results on BAM dataset.

We randomly select and train with 5,000 oil painting images from BAM and test the model on arbitrary oil painting input.

We display the exemplar-based results in Fig. 6. Similar to the results for interior design, the outputs are promising, without any color distortion, and appear to capture the color style of the reference. Also, we show the palette-based results in Fig. 5. As before, we insert diverse color palettes into the model given the same input. The results indicate that ColorHouse is able to translate the various color styles of the given palettes to the inputs.

B.4. Ablation study and Analysis

In this part we conduct analysis for ColorHouse recolorization.

RGB histogram analysis. To better understand how ColorHouse change the RGB value, we conduct an ablation study on RGB histogram. In Fig 7, we show an example of exemplar-based ColorHouse recolorization, associated with the RGB histogram of the reference, input and output images. It could be observed that our model is capable of learning the RGB distribution and shift it towards the reference one with a decent output. For example, ColorHouse leverages the blue pixels and make its distribution close to the reference

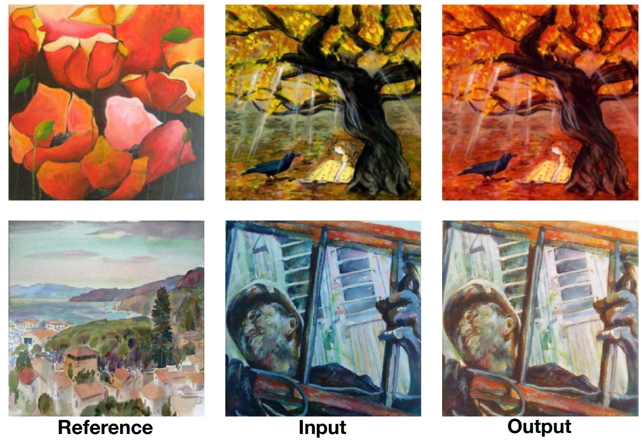


Figure 6: Exemplar-based ColorHouse results on BAM dataset.

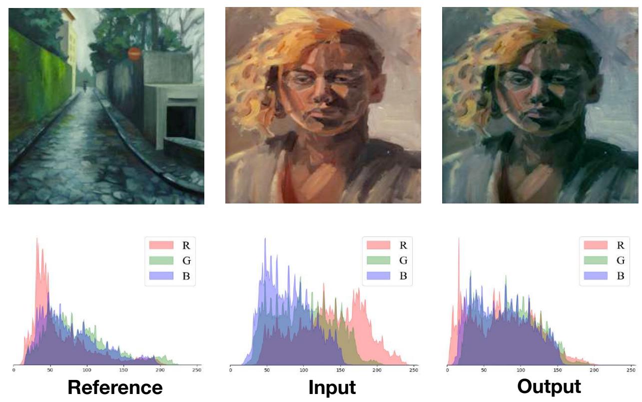


Figure 7: RGB histograms of an exemplar-based ColorHouse result. It could be noticed that our model can learn and leverage the RGB distribution from the reference image to the input.

image.

Illumination. Beyond, we apply illumination adjustment to the generated output for color palette-based ColorHouse. We display the results in Fig 8, it could be noticed that ColorHouse could adjust the illumination to the required value.

B.5. Discussion and Future Work

The problem of recolorization is highly underconstrained, and suitable priors for solving it are often highly contextual. However, in this work we show that, at least for images from design-oriented domains, it is possible to explore a more unconstrained space of plausible alternative colorizations than seen in previous work. This could be used for creative exploration, data augmentation, and other applications as well.

Limitations. Ideally, one could perform recolorization

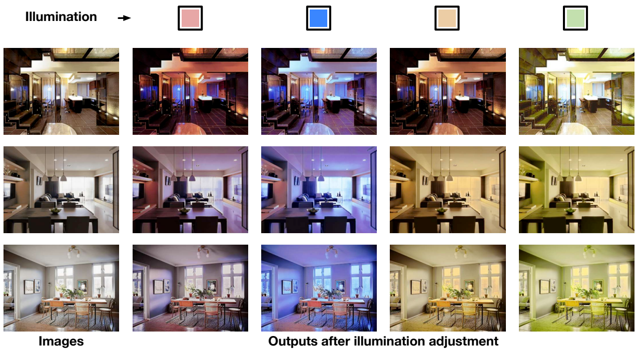


Figure 8: Illumination adaptation. ColorHouse conduct illumination adjustment on the images based on different inserted illumination value.

on arbitrary images that would create reasonable palettes where none may have existed—effectively applying designed aesthetics to random images. This would require not only broader and more extensive training, but also some strategy for learning how distinct colors in arbitrary scenes might be regrouped to form a palette. This is an exciting but difficult direction for future work. A related problem that can be seen in our current results has to do with conflict between the palette loss and the adversarial loss—namely, it is difficult to find plausible recolorings of an image that result in a larger variety of colors than the image started with. For this reason, we often see that inputs with monochrome color harmonies palettes produce outputs that appear to have changed illumination. Our illumination loss helps with this significantly, but is not able to find creative ways to introduce additional colors. This would also be an interesting direction for future work.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014. 1

[2] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):47, 2018. 2

[3] János Schanda. *Colorimetry: understanding the CIE system*. John Wiley & Sons, 2007. 2

[4] Maureen Stone. *A field guide to digital color*. CRC Press, 2016. 2

[5] Michael J. Wilber, Chen Fang, Hailin Jin, Aaron Hertzmann, John Collomosse, and Serge Belongie. Bam! the behance artistic media dataset for recognition beyond photography. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[6] Qing Zhang, Chunxia Xiao, Hanqiu Sun, and Feng Tang. Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing*, 26(4):1952–1964, 2017. 2

[7] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2

[8] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 1