

Color2Embed: Fast Exemplar-Based Image Colorization using Color Embeddings

Hengyuan Zhao^{1*} Wenhao Wu^{1*} Yihao Liu^{2*} Dongliang He¹

¹Department of Computer Vision Technology (VIS), Baidu Inc.

²University of Chinese Academy of Sciences

Abstract

In this paper, we present a fast exemplar-based image colorization approach using color embeddings named Color2Embed. Generally, due to the difficulty of obtaining input and ground truth image pairs, it is hard to train a exemplar-based colorization model with unsupervised and unpaired training manner. Current algorithms usually strive to achieve two procedures: i) retrieving a large number of reference images with high similarity for preparing training dataset, which is inevitably time-consuming and tedious; ii) designing complicated modules to transfer the colors of the reference image to the target image, by calculating and leveraging the deep semantic correspondence between them (e.g., non-local operation), which is computationally expensive during testing. Contrary to the previous methods, we adopt a self-augmented self-reference learning scheme, where the reference image is generated by graphical transformations from the original colorful one whereby the training can be formulated in a paired manner. Second, in order to reduce the process time, our method explicitly extracts the color embeddings and exploits a progressive style feature Transformation network, which injects the color embeddings into the reconstruction of the final image. Such design is much more lightweight and intelligible, achieving appealing performance with fast processing speed.

1. Introduction

Image colorization aims to add vivid colors to a grayscale image for more visually pleasing appearance. As a classic vision task, image colorization has a wide range of applications, such as colorizing old photos, remastering legacy movies and transferring colors between paintings. In the past, there have produced large amounts of black-and-white photos and movies, and colorization can assist the users to give vibrant colors to these old image media, mak-

*Co-first author. This work was done when Hengyuan Zhao was a research intern at Baidu VIS.

ing the elapsed frozen moments relive.

In recent years, deep-learning-based colorization methods have been introduced to solve this problem. There are three types of solutions: fully automatic colorization, user-guided scribble-based colorization and exemplar-based colorization. For fully automatic colorization, existing algorithms [48, 17, 24, 38] have been proposed to map the grayscale image to its color version with the supervised training. Though these methods could directly generate a colorful image without any user intervention, the colorization results are not always satisfactory.

For scribble-based colorization, it utilizes the user-selected color scribbles to guide the production of colorization. The drawback of it is that users are required with extra efforts to select abundant points in specific locations of the given grayscale image, which involves laborious and delicate work. Inappropriate scribbles and locations will lead to artifacts in the final results. Moreover, users are also required for a sense of aesthetics to choose proper and vivid colors from a palette. It is hard for an untrained user to colorize a large collection of images satisfactorily.

As for exemplar-based colorization, the performance is better than that of automatic colorization by adopting the additional information. However, there are two challenges for implementing exemplar-based colorization. First one is how to efficiently prepare the reference database. As the database used in image colorization is always huge (e.g., ImageNet [6].), it is costly to retrieve a relevant and proper reference image for every input image. The second one is that, for an input grayscale image, there lacks its ground truth color image to construct the training pair for supervised learning. In [12], the authors use a pre-trained gray image retrieval algorithm to construct a large reference image database. Obviously, it is extremely time-consuming and cumbersome. Moreover, if we want to conduct experiments on other datasets, the previous retrieval algorithm has to be retrained. As for network structure, most existing methods [12, 29, 16, 46, 47] explicitly compute the semantic correspondence between reference and target images with non-local operations, which will bring about extra

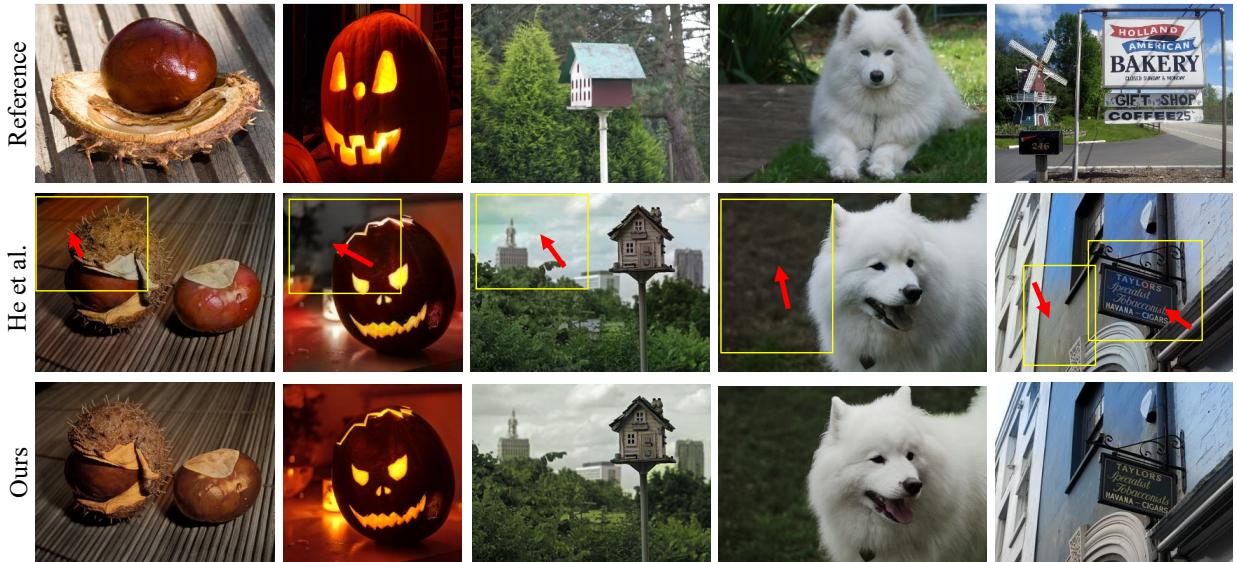


Figure 1. **First row:** reference images. **Second row:** [12] could produce incorrect colors and color contaminations. **Third row:** Without explicit corresponding search, our method can avoid extra artifacts and achieve real-time processing speed with more natural visual results.

computational expensive calculations and could lead to artifacts as shown in Figure 1.

To simplify the implementation of training such a colorization model, we use thin plate splines (TPS) [8, 5] transformation to create several self-reference images from a given color image, so that the training approach can be carried out in a paired supervised manner. Furthermore, our solution does not require any sophisticated or time-consuming processes such as the computation of correspondence maps. To achieve colorization, we extract the color information from the reference color image and then inject the color embeddings into the deep representations of the input grayscale image. Our approach, with such an efficient architecture, can achieve fast processing speed. Different from [29, 46, 25] which adopt a bundle of loss functions, we only use the simplest pixel-wise reconstruction loss [15] and perceptual loss [20] to regularize the learning procedure. These simple losses also reduce the difficulty of reproduction.

The proposed method consists of three main modules: color embeddings generation network, content embeddings generation network and Progressive Feature Formalisation Network (PFFN). The first network extracts high-dimensional feature maps from the reference image in RGB space, after which the deep feature maps are passed through a designed multi-layer perceptron (MLP) [10] to generate the abstract embeddings. The second is devised to encode the target grayscale image into intermediate deep features. The last network takes the content and color embeddings as the input to generate a color image with the collaboration of several Progressive Feature Formalisation Blocks (PFFB), which is proposed to inject the reference color embeddings into the reconstruction process.

2. Related Work

2.1. Fully Automatic Colorization

Learning-based colorization methods [3, 7, 24, 48, 17, 38, 50, 49, 19, 46, 12, 29, 44, 26, 43, 22] have addressed more attention in recent years. Rely on the strong representation ability of deep neural networks, fully automatic colorization methods [3, 7, 24, 48, 17, 38, 50] can be realized by designing various network structures and leveraging a large-scale image dataset. Iizuka et al. [17] proposes a two-branch learning architecture to combine the global priors and local features. Zhang et al. [48] proposes a VGG-like deep network to output regression results and color distributions. Larsson et al. [24] adopts the color histograms for colorization. These methods sensitively suffer from visual artifacts when the input image has complex content with multiple objects. After training, the parameters of the network would be fixed, the output colorization results are only one plausible solution without multimodality

2.2. Scribble-based Colorization

Using human intervention, some algorithms [49, 27, 14, 45, 33, 30, 39] are introduced to propagate the initial color points or strokes to the entire grayscale image. The propagation is based on optimization strategy and pixel similarity metrics. In [27], Levin et al. adopts Markov Random Field for propagating the sparse scribble colors to the adjacent pixels which have similar luminance intensity. Qu et al. [33] and Luan et al. [30] take further consideration of image textures for propagation. Yatziv et al. [45] utilizes the intrinsic distance for constraint. Huang et al. [14] involve the edges to avoid the color blending for advanced color propagation. As for scribble-based colorization method, Zhang et al. [49] proposes a U-net-like network to interactively

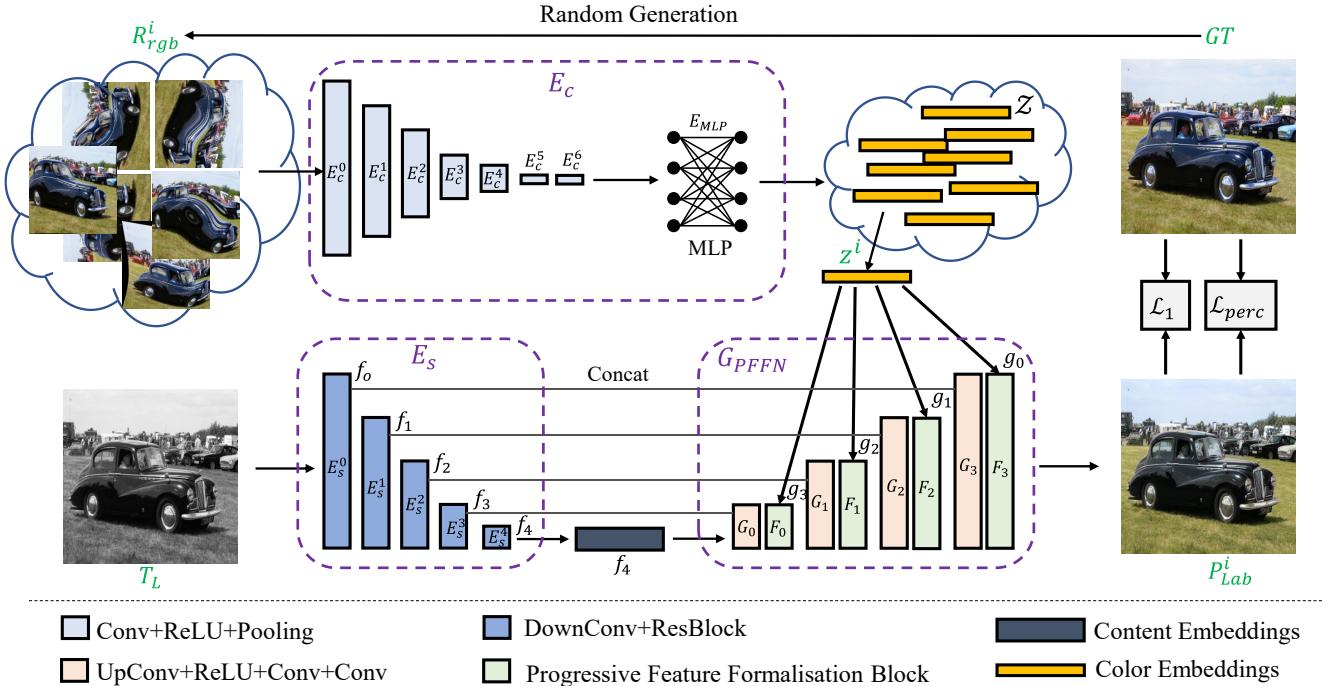


Figure 2. An illustration of the proposed Color2Embed framework, which encodes the reference images into color embeddings for exemplar-based image colorization.

produce impressive results with sparse color hints and color histograms. These scribble-based colorization methods are prone to be restricted by the aesthetic of users. It is hard for an untrained user to select suitable points and select correlated colors from a palette.

2.3. Exemplar-based Colorization

Compared to scribble-based methods, exemplar-based colorization only requires the user to select a suitable reference image according to the target grayscale image. Earlier traditional works [42, 34] transfer colors by matching the global color statistics. More early traditional methods [18, 40, 2, 9, 4, 28, 1] focus on adopting techniques of extracting local image features like segmented region [18, 40, 2], super-pixel [9, 4], and pixel-level [28, 1]. But these methods are vulnerable to be affected by uncorrelated regions in an image and always bring unexpected redundant outlines of objects. A pioneer work [12] first proposes a deep exemplar-based colorization algorithm that utilizes the pre-trained VGG-19 to extract deep features for similarity calculation and then warps reference image. Xiao et al. [43] proposes a pyramid structure network to exploit the inherent multi-scale color representations. Lu et al. [29] proposes an approach that jointly considers the semantic correspondence and global color histograms into network design. Xu et al. [44] proposes a fast deep exemplar-based method by adopting an AdaIN-based [13] transfer network and randomly generates color hints of the transferred image. Such random sampling procedure prone to produce unstable

colors by the incorrect selection of color hints.

3. Method

3.1. Overview

Following previous methods [48, 49, 46, 12, 29, 44, 43, 17], we perform this task in CIE Lab [36] color space. Each image can be separated into a luminance channel L and two chrominance channels a and b . Given an $H \times W$ grayscale target image $T_L \in R^{H \times W \times 1}$ containing only the luminance channel and a color reference image $R_{rgb} \in R^{H \times W \times 3}$ which is represented in RGB space, the aim of exemplar-based colorization is to find a function $F(T_L|R_{rgb})$ that predicts the corresponding a and b channels P_{ab} . Different from previous methods, we directly extract the color embeddings in RGB color space while our reconstruction process is performed in Lab color space.

The overall structure of the proposed end-to-end network is shown in Figure 2. E_c , E_s and G_{PFFN} are color encoder, content encoder and Progressive Feature Formalisation Networks (PFFN) respectively. E_c takes the random reference R_{rgb}^i as input and generates the color embeddings z^i . We denote Z as the set of all color embeddings.

$$z^i = E_c(R_{rgb}^i). \quad (1)$$

E_s represents the content encoder network which contains several downsampling convolutions and resblocks to extract the intermediate features and content embeddings.

$$f_N = E_s^{N-1}(f_{N-1}). \quad (2)$$

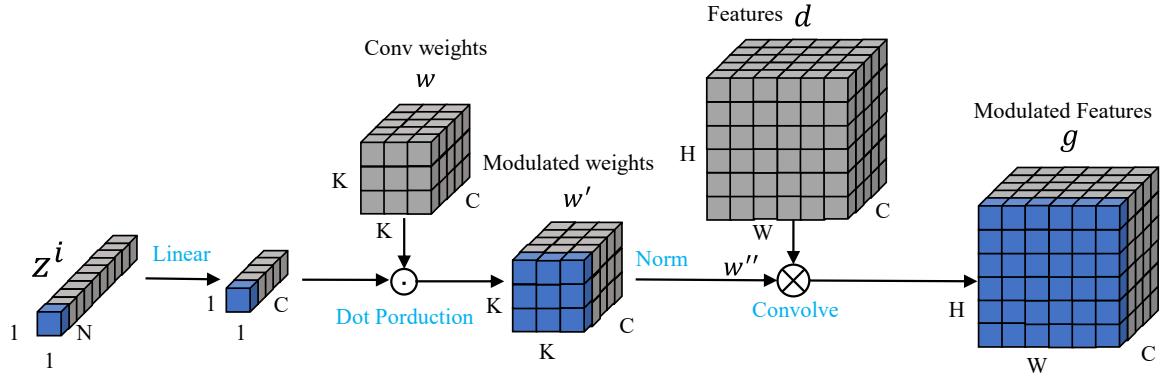


Figure 3. The illustration of progressive feature formalisation block.

These intermediate features are passed to G_{PFFN} for providing multi-scale content information. G_{PFFN} contains consecutive upsampling blocks and Progressive Feature Formalisation Blocks (PFFB).

$$g_i = \begin{cases} F_i(G_i(f_N, f_{N-1})), & \text{if } i = N - 1, \\ F_i(G_i(g_{i+1}, f_i)), & \text{otherwise,} \end{cases} \quad (3)$$

where $i \in \{1, \dots, N - 1\}$, G_i represents the upsampling module and F_i represents the PFFB. For easily implementation, we only adopt two wildly used loss functions, reconstruction loss and perceptual loss. Remarkably, during training, we generate R_{rgb} from ground truth by applying the Thin Plate Splines (TPS) transformation [8, 5]. During testing, our model can accept any given color images as references to colorize the grayscale image.

In the following, we will introduce Self-Augmented Self-Reference Learning in Sec. 3.2, PFFN in Sec. 3.3, and loss functions in Sec. 3.4.

3.2. Self-Augmented Self-Reference Learning

As preparing a reference image for a grayscale image and conjugating these two inputs for pixel-wise paired training is an obstacle, we adopt the reference generation method from [25]. To generate random reference image R_{rgb}^i for a given target image T_L , we apply spatial transformations on ground truth image GT . Since R_{rgb}^i is essentially generated from GT , these processes guarantee the sufficient color information to colorize T_L , which encourage the proposed framework to reflect the R_{rgb}^i in the colorization process. The detail on how these transformations operate is described in the following. First, the content transformation $C(\cdot)$ adds a particular random noise on GT . The reason why we impose the noise of the original GT is to increase the training samples. The same ground truth GT could have various reference images. Afterwards, we further apply TPS transformation $T(\cdot)$, a non-linear spatial transformation on $C(GT)$, resulting in our final reference image R_{rgb}^i . This prevents our model from lazily bringing the color in the same pixel location from GT while enforcing the color

Encoder E_c to only extract the semantic color information from a reference image even with a spatially different layout. The above two transformations help our model learn to transfer color information from an exemplar image to a target image and avoid learning an identity mapping.

Discussion. In the proposed method, we first encode the reference image into a lower-dimensional embeddings which is supposed to represent the color information. Why does such mechanism work? How to ensure that the extracted embeddings contain only color information without image content information? As shown in Figure 2, during training, given an original reference color image GT , we can obtain multiple reference versions R_{rgb} with different geometric transformations, which are not aligned in image content but contain coherent color characteristics. More importantly, these generated reference images are not aligned with the target image T_L in pixel-level as well. They are demanded to only provide the color guidance for T_L . Thus, the supervision signal only acts on the reconstruction of color without consideration of the image content. Extensive experiments have also demonstrated the effectiveness of injecting such color embeddings for obtaining exemplar-based colorization results.

3.3. Progressive Feature Formalisation Network

To formalize the content features, we embed the feature modulation module from StyleGAN2 [21] into our PFFN. As shown in Figure 3, the convolution weights w is the initial trainable weights, and we first scale it with a constant hyperparameter s corresponding to the channel dimension of the input feature maps and project it with a linear operation to align the channel size. It can be formulated by

$$w' = w \cdot s \cdot F_{Linear}(z), \quad (4)$$

where w and $w' \in R^{C_i \times C_j \times K \times K}$ are the original and modulated convolution weights, respectively. The kernel size of convolution weights is K while C_i and C_j represent the number of input and output channels. The F_{Linear} transfers the color embeddings z^i to modulate the weights w in

this feature scale. After the modulation, we adopt the normalization procedure with F_{Norm} to further constrain the values of convolution weights. The F_{Norm} can be formulated as

$$F_{Norm} = \frac{1}{\sqrt{\sum w'^2 + \epsilon}}, \quad (5)$$

where the ϵ is a small constant to avoid the numerical issues. As for the detail of F_{Norm} , we normalize the dimension of w' except from the input feature dimension. This operation can be regarded as a kind of vector unitization, making the convolution weights focus more on the direction rather than the absolute values. From our experience, this operation also stabilizes the output performance during training. As depicted in Figure 3, F_{Linear} and F_{Norm} represent ("Linear") and (operations, respectively. The final convolution weights is w'' , which can be formally written by

$$w'' = F_{Norm}(w'). \quad (6)$$

Given a content features d , the modulated features g can be formulated by

$$g = F_{convolve}(w'', d), \quad (7)$$

where $F_{convolve}$ represents convolution operation.

3.4. Loss Function

Most existing methods tend to devise and utilize a variety of complex loss functions to achieve multiple constraints in their design. Ascribing to the effective and ingenious design of our method, we only need to adopt two widely-used loss functions to accomplish satisfactory colorization performance.

Reconstruction Loss. Owing to the self-augmented self-reference training mechanism, it is able to conduct paired supervised learning, which can make the training process more stable and accelerate the convergence. We adopt smooth $L1$ loss [15] as the distance metric to avoid the averaging solution in the ambiguous colorization problem [49]. The reconstruction loss can be formulated as:

$$\mathcal{L}_{recon} = \sum_x \text{smoothL1}(P_{ab}^i(x), GT_{ab}(x)), \quad (8)$$

where $\text{smoothL1}(a, b) = \frac{1}{2}(a - b)^2, if |a - b| < 1, \text{smoothL1}(a, b) = |a - b| - \frac{1}{2}, otherwise.$

Perceptual Loss. To allow the network predict the perceptually plausible colors even without a proper reference image, we adopt a perceptual loss [20] to constrain the predicted P_{rgb}^i and GT . The P_{rgb}^i is transformed from P_{Lab}^i with a color space transformation and P_{Lab}^i is concatenated from input grayscale image T_L and output P_{ab}^i . Formally,

$$\mathcal{L}_{perc} = \sum_x \|F_P(x) - F_{GT}(x)\|_1, \quad (9)$$

where F_P and F_{GT} represent the feature maps extracted from VGG-19 $relu5_2$ layer from P_{rgb}^i and GT , respectively.

In summary, the overall loss function for training is defined as:

$$\mathcal{L}_{total} = \lambda_{rec}\mathcal{L}_{recon} + \lambda_{perc}\mathcal{L}_{perc}, \quad (10)$$

where the λ_{rec} and λ_{perc} represent the weights of the two losses, respectively.

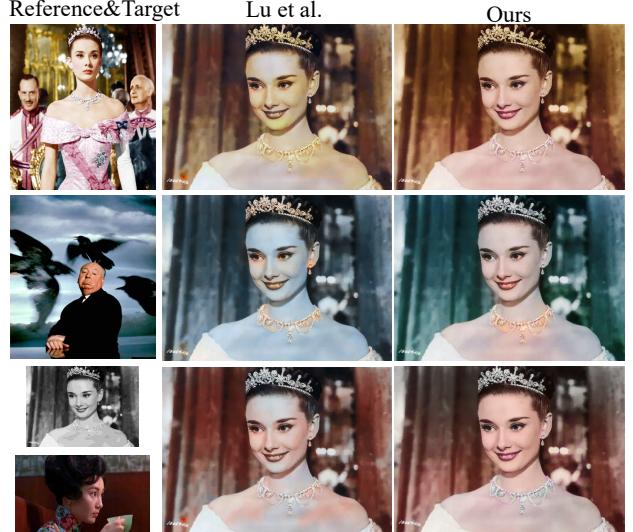


Figure 4. Comparison with Lu et al. [29] on Image *Hepburn*.

4. Experiments

4.1. Implementation Detail

Dataset and Metric. ImageNet[6] is a commonly used image classification dataset with 1.2 million images from 1000 categories. Following previous image colorization methods [48, 49, 12, 29], we adopt this dataset as our ground truth color image dataset to generate reference images and grayscale images. We resize all training images into size 256×256 with bilinear rescaling method. During generating reference image, we add random Gaussian noise with mean 0 and variance $\sigma = 5$. The geometric transformation of TPS is randomly applied, hence, the same image has various transformed pairs every time. For data augmentation, we randomly rotate and flip the reference image. As for test datasets, we use the images from He et al. [12] for a fair comparison.

Training Details. The H and W is 256, C are both 512. The size of color embeddings z^i is $512 \times 1 \times 1$. When training, we use PyTorch [32] framework to train our model with batch size 64 on 8 NVIDIA 1080Ti GPUs. The learning rate is 1×10^{-4} using the Adam [23] opti-



Figure 5. Comparison with state-of-the-art exemplar-based colorization methods. The first two rows are input target-reference image pairs. The last six rows are corresponding colorization results generate by [1, 9, 11, 29, 44] and the proposed method, respectively. **Red rectangles** highlight failures and artifacts. Please zoom in for best view.

mizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We set $\epsilon = 1 \times 10^{-8}$, $\lambda_{rec} = 1$, $\lambda_{perc} = 0.1$.

4.2. Comparison with State-of-the-art Methods

To evaluate the effectiveness of our method, we compare our method with five state-of-the-art methods: Gupta et al. [9], Bugeau et al. [1], He et al. [12], Lu et al. [29], Xu et al. [44]. These methods are all exemplar-based colorization approaches, where [9] and [1] are traditional methods, [12, 29, 44] are deep learning-based methods. To provide a fair comparison, we directly borrow their released results or run their test code to generate results.

Qualitative Evaluation. To compare with the existing exemplar-based methods, we run our algorithm on 35 pairs collected from [12]. Figure 5 shows some representatives. The traditional methods [9] and [1] bring about many artifacts in the first image. [9] leads to unbalanced colors in

all five images and [1] has unexpected colors or redundant outlines of objects. [12] tends to bring color contamination at image local regions like in the first, second, fourth, and fifth images. [29] is a newly proposed method that first incorporates the local semantic correspondence and global color histogram for consideration to generate the colorized image. In Figure 5, due to the unstable influence of semantic correspondence, it also brings about color contamination in the first and fifth images. To reveal the advantage of our approach, we compare with two methods with explicit computation of correspondence map in Figure 1 and Figure 4.

Robustness of diverse references. In order to evaluate the robustness of diverse references, we additionally provide the qualitative comparison on five reference images in Figure 6. As mentioned in Figure 1, [12] and [29] tend to bring unexpected artifacts due to the incorrect computation of correspondence map.



Figure 6. Results of colorization on different reference images. When the reference image content is not similar with target image, He et al. [12] and Lu et al. [29] produce more artifacts due to its explicit computation of correspondence map.



Figure 7. Results of legacy pictures. In each set, the target image, the reference and ours result lie on the left, middle, and right, respectively.

Results on old photos. We further evaluate the performance of Color2Embed on real old photos collected from the internet, as shown in Figure 7 and Figure 9.

Table 1. The running time (second) comparison of exemplar-based colorization methods. OOM denotes out of memory. All the experiments are tested on single NVIDIA 1080Ti GPU.

Image Size	256×256	512×512	1024×1024
He et al. [12]	7.25+1.14s	33.69+1.30s	49.96+OOM
Xiao et al. [43]	0.48s	1.4s	3.96s
Lu et al. [29]	0.44s	1.37s	OOM
Xu et al. [44]	0.04+0.08×2s	0.14+0.12×2s	0.59+0.28×2s
Ours	0.03s	0.08s	0.32s

Running Time. In this study, we focus on providing a

fast algorithm by removing the computation of correspondence map. We compare with other four exemplar-based colorization methods to illustrate our advantage over running time in Table 1. Notice that He et al. [12] and Xu et al. [44] have two stages: color transfer stage and colorization stage. They cost much more time than others.

5. Ablation Study

Effectiveness of Self-Augmented Self-Reference

Learning. To evaluate the effectiveness of self-augmented self-reference learning, we train Color2Embed directly with the ground truth as reference without self-augmented operation. Figure 8 (a) appears obvious color contamination since the model learns to retain positions of these colors in the reference image. On the contrary, Figure 8

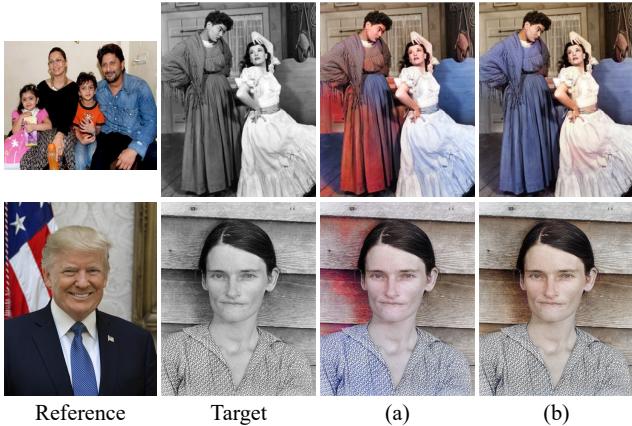


Figure 8. Effectiveness of self-augmented self-reference learning.
(a) without self-augmentation. (b) with self-augmentation.



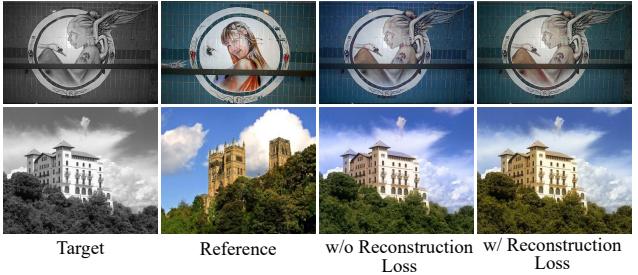
Figure 9. Different reference images will change the color style in the results. Our method has the robustness of accepting diverse references.

(b) successfully extracts colors from reference and then propagates them to appropriate regions.

Effectiveness of Loss functions. To evaluate the effectiveness of loss functions, we train Color2Embed with single reconstruction or perceptual loss, respectively in Figure 10(a) and 10(b). When train Color2Embed without perceptual loss, the model will produce unsatisfactory results as shown in Figure 10(a). The second row of dog, car and



(a) Four examples for evaluating the effectiveness of perceptual loss.



(b) The effectiveness of reconstruction loss.

Figure 10. Effectiveness of loss functions. Only one single loss leads to unsatisfactory performance and accurate colors are observed by combining these two losses.

monkey images obtain incorrect colors. We demonstrate the effectiveness of reconstruction loss will influence the coherence between target and reference images. In the first row of Figure 10(b), the third image produce unseen colors of background which is really different with reference. In the second row, reference image shows a yellow tower, while the building of third image shows white color.

6. Conclusion

In this paper, we present a fast exemplar-based colorization algorithm named Color2Embed. Our method first converts the reference color images into color embeddings and then utilize the extracted color embeddings to generate robust results by the proposed progressive feature formalisation network. The variety of training pairs ensure the color embeddings generation network only extract color information from reference. This mechanism makes our approach have the ability of receiving any color images as reference without bringing severe artifacts. Furthermore, we typically find that our method does no need to adopt excess loss functions compared to existing methods. Extensive experiments show that our results surpass other state-of-art methods in qualitative comparison and running time.

References

- [1] Aurélie Bugeau, Vinh-Thong Ta, and Nicolas Papadakis. Variational exemplar-based image colorization. *IEEE Transactions on Image Processing*, 23(1):298–307, 2013.
- [2] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. In *European conference on computer vision*, pages 126–139. Springer, 2008.
- [3] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.
- [4] Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin. Semantic colorization with internet images. *ACM Transactions on Graphics (TOG)*, 30(6):1–8, 2011.
- [5] Haili Chui and Anand Rangarajan. A new algorithm for non-rigid point matching. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 44–51. IEEE, 2000.
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [7] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 567–575, 2015.
- [8] Jean Duchon. Splines minimizing rotation-invariant seminorms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.
- [9] Raj Kumar Gupta, Alex Yong-Sang Chia, Deepu Rajan, Ee Sin Ng, and Huang Zhiyong. Image colorization using similar images. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 369–378, 2012.
- [10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [12] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):1–16, 2018.
- [13] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.
- [14] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on MultiMedia*, pages 351–354, 2005.
- [15] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.
- [16] Satoshi Iizuka and Edgar Simo-Serra. Deepremaster: temporal source-reference attention networks for comprehensive video enhancement. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.
- [17] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016.
- [18] Revital Ironi, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. In *Rendering techniques*, pages 201–210. Citeseer, 2005.
- [19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
- [21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.
- [22] Siavash Khodadadeh, Saeid Motiian, Zhe Lin, Ladislau Boloni, and Shabnam Ghadar. Automatic object recoloring using adversarial learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1488–1496, January 2021.
- [23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [24] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016.
- [25] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5801–5810, 2020.
- [26] Chenyang Lei and Qifeng Chen. Fully automatic video colorization with self-regularization and diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3753–3761, 2019.

- [27] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694. 2004.
- [28] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. Intrinsic colorization. In *ACM SIGGRAPH Asia 2008 papers*, pages 1–9. 2008.
- [29] Peng Lu, Jinbei Yu, Xujun Peng, Zhaoran Zhao, and Xiaojie Wang. Gray2colornet: Transfer more colors from reference image. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3210–3218, 2020.
- [30] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320, 2007.
- [31] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013.
- [32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [33] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. *ACM Transactions on Graphics (TOG)*, 25(3):1214–1220, 2006.
- [34] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.
- [35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [36] Janos Schanda. Cie colorimetry. *Colorimetry: Understanding the CIE system*, pages 25–78, 2007.
- [37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [38] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7968–7977, 2020.
- [39] Daniel Sýkora, John Dingiana, and Steven Collins. Lazy-brush: Flexible painting tool for hand-drawn cartoons. In *Computer Graphics Forum*, volume 28, pages 599–608. Wiley Online Library, 2009.
- [40] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. Local color transfer via probabilistic segmentation by expectation–maximization. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 747–754. IEEE, 2005.
- [41] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.
- [42] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 277–280, 2002.
- [43] Chufeng Xiao, Chu Han, Zhuming Zhang, Jing Qin, Tien-Tsin Wong, Guoqiang Han, and Shengfeng He. Example-based colourization via dense encoding pyramids. In *Computer Graphics Forum*, volume 39, pages 20–33. Wiley Online Library, 2020.
- [44] Zhongyou Xu, Tingting Wang, Faming Fang, Yun Sheng, and Guixu Zhang. Stylization-based architecture for fast deep exemplar colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9363–9372, 2020.
- [45] Liron Yatziv and Guillermo Sapiro. Fast image and video colorization using chrominance blending. *IEEE transactions on image processing*, 15(5):1120–1129, 2006.
- [46] Bo Zhang, Mingming He, Jing Liao, Pedro V Sander, Lu Yuan, Amine Bermak, and Dong Chen. Deep exemplar-based video colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8052–8061, 2019.
- [47] Qian Zhang, Bo Wang, Wei Wen, Hai Li, and Junhui Liu. Line art correlation matching feature transfer network for automatic animation colorization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3872–3881, January 2021.
- [48] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.
- [49] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.
- [50] Jiaojiao Zhao, Li Liu, Cees GM Snoek, Jungong Han, and Ling Shao. Pixel-level semantics guided image colorization. *arXiv preprint arXiv:1808.01597*, 2018.