DOI: 10.1111/cgf.14950

Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

Palette-Based and Harmony-Guided Colorization for Vector Icons

Miao Lin[1†] I-Chao Shen[2†‡] Hsiao-Yuan Chin[1] Ruo-Xi Chen[1] Bing-Yu Chen[1]

[1] {r10725041, b08207054, r11922018, robin}@ntu.edu.tw, National Taiwan University, Taiwan
[2] ichaoshen@g.ecc.u-tokyo.ac.jp, The University of Tokyo, Japan

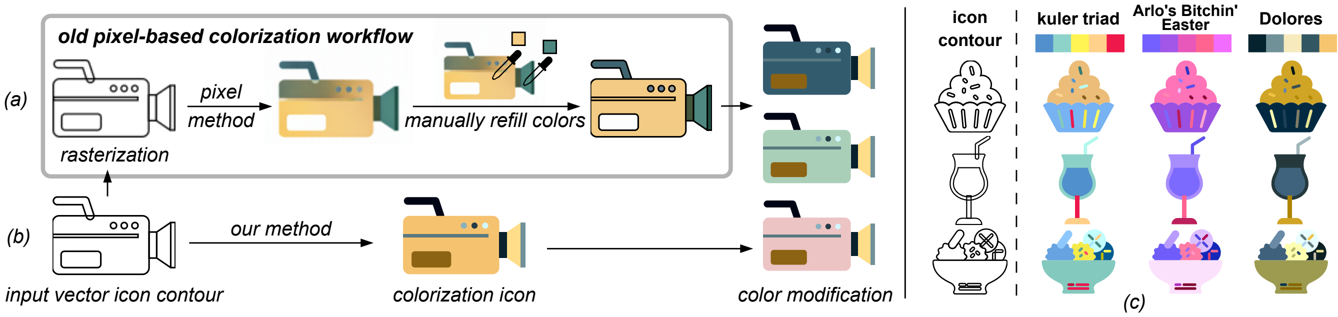

Figure 1: *(a) Previous studies on pixel icon colorization [LLW22, SLWW19] require additional manual color refillment to further edit the colorization results. (b) In contrast, our method allows designers to seamlessly colorize the input vector icon and modify the colorization results. (c) Our method allows designers to efficiently colorize icons within a set and evaluate the suitability of the chosen color palette.*

Abstract

Colorizing icon is a challenging task, even for skillful artists, as it involves balancing aesthetics and practical considerations. Prior works have primarily focused on colorizing pixel-based icons, which do not seamlessly integrate into the current vector-based icon design workflow. In this paper, we propose a palette-based colorization algorithm for vector icons without the need for rasterization. Our algorithm takes a vector icon and a five-color palette as input and generates various colorized results for designers to choose from. Inspired by the common icon design workflow, we developed our algorithm to consist of two steps: generating a colorization template and performing the palette-based color transfer. To generate the colorization templates, we introduce a novel vector icon colorization model that employs an MRF-based loss and a color harmony loss. The color harmony loss encourages the alignment of the resulting color template with widely used harmony templates. We then map the predicted colorization template to chroma-like palette colors to obtain diverse colorization results. We compare our results with those generated by previous pixel-based icon colorization methods and validate the effectiveness of our algorithm by evaluations in both qualitative and quantitative measurements. Our method enables icon designers to explore diverse colorization results for a single icon using different color palettes while also efficiently evaluating the suitability of a color palette for a set of icons.

1. Introduction

With the widespread adoption of the digital technologies, including computers, intelligent appliances, and wearable devices, the role of icons has become increasingly important. These icons possess the ability to bridge the language barrier, allowing people who speak different language to comprehend their meaning without the need for additional text description. However, creating icons with visually captivating expressions requires proficient design skills. A crucial aspect of icon design is the colorization process. Designers commonly need to carefully consider a range of color references, such as text descriptions, color palettes, reference images, and so on. Furthermore, designers must approach each design case-by-case because different contexts may requires different icon styles. This process can be time-consuming and labor-intensive.

Previously, various neural-based methods have been proposed to simplify the colorization process of pixel icons. For instance, Sun *et al.* [SLWW19] proposed a dual conditional GAN that si-

† Both authors contributed equally to the paper.
‡ Corresponding author

multaneously regulates the icon contours from the content input and the color combinations from the color reference icons. Li *et al.* [LLW22] employed an encoder-decoder network for icon colorization and used conditional normalizing flow to enable the network to generate different results based on user-specified styles. Although these approaches have achieved impressive icon colorization results, they are limited to pixel icons. Therefore, if designers wish to modify the icon shape or colors after the colorization process, they need to manually refill the colorization results back to the icon contours for further editing (Figure 1(a)), which is a time-consuming and labor-intensive task. As a result, most previous works for colorizing pixel icons cannot be seamlessly integrated into the standard icon design workflow.

In this paper, we propose a palette-based colorization method for vector icons directly. Our method takes an input icon in scalable vector graphics (SVG) format and a five-color palette. The goal is to predict harmonious colors for each path in the input icon. Our method involves two main steps: generating a colorization template and performing palette-based color transfer. To generate the colorization template, we propose a vector icon colorization network trained using a novel combination of a MRF-based loss and a color harmonic loss functions. This network generates a colorization template that captures the likelihood of colorization for each path, ensuring the resulting color combinations align with human perception of color harmony. In the final step, we sample colorization candidates from the colorization template and transfer color from the palette to the icon by optimizing the relative luminance between paths. It is important to note that we choose to use a five-color palette based on the interview results with the icon designers. Our color transfer step operates independently of the colorization template generation, enabling it to accommodate arbitrary numbers of colors in the color palette.

We conducted a user study to evaluate the results obtained by our method and baseline methods. The study result demonstrates that the results of our method obtained the highest score compared to other baseline methods. In the feedback provided by the participants, it was noted that our colorization results were preferred due to the clarity of the icon and the ease of distinguishing the foreground and background. These findings highlight the major advantage of our vector icon colorization method. Moreover, the seamlessly integration into the standard icon design pipeline providing designers two key benefits. First, they can explore a diverse range of colorization results for a single icon using different color palettes. Meanwhile, they can efficiently evaluate the suitability of a color palette for a set of icons.

The major contributions and novelties of this paper include:

- To the best of our knowledge, we present the first vector icon colorization method that directly predicts harmonious colors for each path.
- We propose a novel MRF-based loss and a color harmony loss to train the vector icon colorization network.
- We conducted a user study to demonstrate the benefit of our method operating directly on vector icon.

2. Related Work

2.1. Color Harmony Evaluation

The evaluation of color aesthetics is an essential subject with a long history. Jacobson and Ostwald [JO48] considered that harmony equals order. Moon and Spencer [MS44] built a harmony classification. Itten [Itt70] drew up basic color theory and developed applications of the color wheel. One well-known theory is the harmonic template on hue provided by Matsuda [Mat95] and Tokumaru *et al.* [TMI02]. They devised eight types of harmonic templates based on the color wheel. Each type except type N is defined as one or two contiguous gray blocks. When color combinations fall entirely into the gray area, they are defined as harmonious color combinations. Based on these templates, Cohen-Or *et al.* [COSG*06] proposed a color harmonization method that quantized image colors into histograms and increased image coordination. Zaeimi and Ghoddosian citezaeimi2020color combined these templates and Munsell color system [Coc14] to advance an optimized color harmony algorithm.

In addition to the traditional color theorem, many studies analyze the harmony of colors through model training. Ou and Luo [OL06] investigated the harmony of two colors. They proposed essential factors that affect color harmony, such as any two colors equal in hue and chroma that differ only in luminance are harmonious. Ou *et al.* [OCLM11] predicted the overall harmony of the three-color combination. O'Donovan *et al.* [OAH11] utilized an extensive online dataset to study color compatibility. Kita *et al.* [KM16] inherited this compatibility research and further studied the palette expansion while maintaining color harmony.

Our study uses online icons as target data. Through the designer's works, we aim to extract color-matching principles more suitable for icon coloring. At the same time, we adopt the widely used harmonic templates [Mat95, TMI02] as the basis for judging color blending, encouraging the model to generate harmonious color combinations.

2.2. Reference-Based Colorization

The task of colorizing nature images usually starts with a gray-scale image. Lin *et al.* [LRFH13] used factor graphs for pattern coloring. Kim *et al.* [KYKL14] implemented a rule-driven approach to colorize segmented images. Zhang *et al.* [ZZI*17] developed a user-guided method to lead image colorization. Another common practice is to use CNNs to complete coloring [ZIE16, ISSI16]. Isola *et al.* [IZZE17] further implemented image-to-image translation tasks in various fields.

Unlike images, the initial stages of line art creation are usually not gray-scale images. Stuff like comics, cartoons, and sketches usually only have black lines as input. GAN-based methods are commonly used to generate colors for line characters [HA17, LQLW17]. Some works also use another picture or sketch as a reference [ZJLL17, LKL*20, LGK*22]. These reference images are usually taken from the same category as the input sketch. It is expected that the same parts to get similar color transformations, such as hair color for hair and skin color for skin. Zhang *et al.* [ZLW*18] proposed a semi-automatic method where users only need to give color hints instead of complex images. Ci *et al.* [CMW*18] also allowed users to color the entire picture by drawing simple color lines. Kim *et al.* [KJPY19] did not require the user to choose a color but only to provide simple text prompts.

Icons are also a type of line art. But compared to sketches, they are more minimalist, more abstract, and even smaller. Sun *et al.* [SLWW19] first proposed the work of icon coloring. They used two discriminators to compute the structural and stylistic similarity of content and color reference icons. Han *et al.* [HZZ20] created a GAN network based on three conditions, adding a mask to limit the scope of the icon content to avoid mixing with the background. Li *et al.* [LLW22] adopted an encoder-decoder architecture and applied normalizing flow so that the same style can generate various color combinations. Although these works' results are appealing, since they are based on pixel icons, these methods are challenging to apply directly to vector graphics. Our goal is to extend icon coloring work to vector formats.

2.3. Image Recoloring

We define image recoloring as recoloring from color images, not gray-scale images or sketches. A common practice in natural image recolor work is to add color features from one image to another [RAGS01, TJT05, HSGL11, WYX11]. Some work focused on luminance adjustment [BPD06, LFUS06, BPCD11], some are based on the principle of harmony, ignoring brightness [COSG*06, LZNH15].

In the absence of image references, some works are devoted to extracting the primary colors from the original image and presenting the picture in the form of a palette. Csurka *et al.* [CSMS10] used semantics and annotations for palette classification and color transfer. Yoo *et al.* [YPCL13] used the primary color of the image as the basis for image region segmentation. Wang *et al.* [WJC13] formulated color adjustment as an interpolation problem and adjusted the color of the image through the sentiment words input by the user. Chang *et al.* [CFL*15] computed luminance and *ab* channels separately and corrected for colors outside the visual range. However, our color references are chosen by the user and do not refer to the color style of the original image.

Our requirements are closer to palette-guided recoloring work. Wang *et al.* [WYW*10] performed soft segmentation on the image, and the user can apply local refinement according to the block of interest. Instead of using a color transfer function, Cho *et al.* [CYM-LYC17] obtained realistic recolored images through image feature learning. Zhang *et al.* [ZXST17] regarded each color as a linear combination of some primary colors. Unlike real pictures, the area that icons can effectively segment is relatively small, and the segmented content does not necessarily contain substantive meaning. We refer to the traditional color transformation method proposed by Chang *et al.* [CFL*15]. Our task is palette-driven. We calculate luminance separately from chroma to transform color within icons' simple lines.

2.4. Applications of Vector Graphics

Vector graphics is a sequence of parameters. The advantage is that this format can be scaled infinitely, ensuring the content is not distorted and is more precise than raster images. Zhang *et al.* [ZYZ*17] and Ha and Eck [HE17] first tried to simulate the generation of sketch strokes by sequence modeling. With the introduction of attention mechanism, Carlier *et al.* [CDAT20] and Ribeiro *et al.* [RBCP20] used a transformer-based architecture to learn feature representation and interpolation for vector sketches. Zou *et al.* [ZSQ*21] treated strokes as a vector format and expanded from sketches to generate realistic drawings in vector format. In addition to simulating stroke generation, there is more and more research on how to convert raster graphics directly into vector graphics or the converting between raster and vector space [LHES19, RGLM21, WL21, SC21].

Apart from the synthesis of vector graphics, how to analyze the characteristics of vector graphics is also an essential topic. In recognition of vector graphics, Li *et al.* [LZZ*18] identified the category of the target object by extracting the keystroke features of the input vector sketch. Xu *et al.* [XHY*18] introduced a novel hashing loss for sketch retrieval. Collomosse *et al.* [CBJ19] learned a search embedding that unifies vector and raster representations. Jiang *et al.* [JLS*21] developed a dual-stream GNN architecture that does not require rasterization and realized object detection and classification of vector objects.

This paper refers to the method proposed by Jiang *et al.* [JLS*21] for analyzing vector context to recognize the composition of vector icons. Different from previous studies, the task of pattern recognition is often performed on objects with distinct features. For example, Jiang *et al.* [JLS*21] implemented object detection on floor plans and diagrams with standard format. However, icons are abstract expressions, which means that the elements of an icon do not necessarily have substance. Therefore, in addition to analyzing the independent features of each path element, we add group features to increase the recognition ability of the model for icon representations.

3. Overview

Inspired by the standard icon design workflow, we develop our algorithm to consist of two steps: generating a colorization template (Figure 2) and performing palette-based color transfer (Figure 4). In the template generation phase, we use SVG icons as training input, extracting the graph node (yellow block) and edge features and using the GNN model to predict each path's color. The training flow is considered a classification task. For the predicted colors, we use the unary loss to judge the correctness of the output colors and the pairwise loss to compare the relationship between each path pair (blue blocks). We also use two terms to control the diversity (purple block) and harmony (orange block) of predicted colors, which balance the coordination of the output color combination.

In the color transfer phase, designers are asked to input an SVG icon contour and a five-color palette. Given the selected icon and palette, our colorization model (green block) will output a color template, representing the proper color combination of the input icon. Then, we transfer the chroma and luminance according to the selected palette, filter harmonious combinations and provide diverse colorization results.

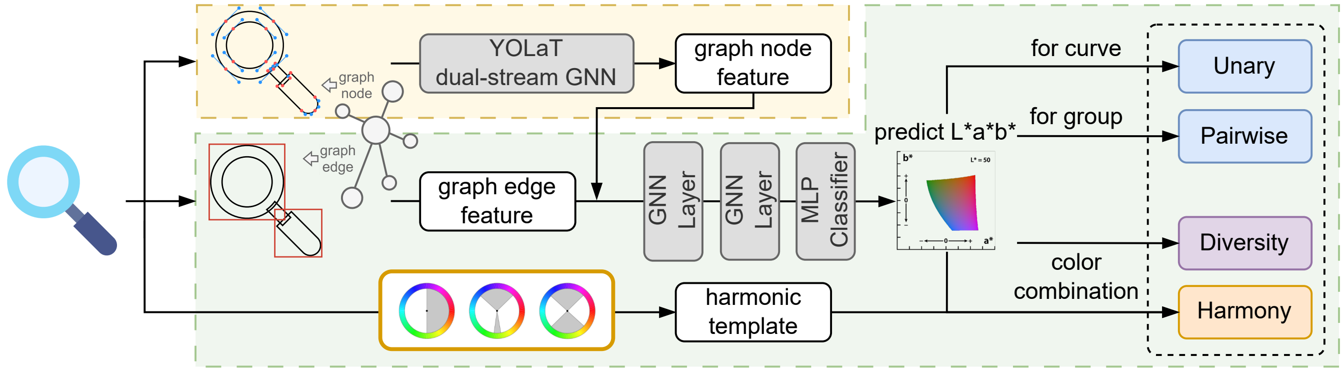

Figure 2: *Given an icon contour, our colorization network extracts unary path features and pairwise features to predict the color template. We propose a novel MRF-based loss that include the unary loss and pairwise loss. Additionally, we introduce a novel color harmony loss and a color diversity loss to enhance the harmonious of the resulting colorizations.*

4. Generating Colorization Templates

Our goal is to generate a colorization template for an SVG icon consisting of N paths represented as $\mathcal{P} = \{p_1, p_2, ..., p_N\}$, where each path consists of a series of cubic Bézier curve. The colorization template is designed to adhere to the principles of color harmony. Specifically, we define the colorization template of $\mathcal{P}$: $\mathcal{L}_{\mathcal{P}} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_N\}$, which comprises probabilities indicating the likelihood of assigning a particular color to each path. To predict the color of each path, we use the CIE *Lab* color space, known for its close approximation to human visual perception. To avoid generating identical colors for all paths, we adopt the technique used in Zhang *et al.* [ZIE16]. Specifically, we approach the problem as a classification task, separately predicting L and ab values. We discretize L, ab into ten bins each, resulting in L having 10 classes, ab having 26 classes each. To ensure that the predicted colors fall within reasonable regions, we treat ab as a two-dimensional plane with 26×26 classes, accounting for different chroma ranges displayed at various luminance levels.

4.1. Data Preprocessing

Given an SVG icon, we normalize the icon size to 512×512. We represent the input vector $\mathcal{P}$ as an icon graph $G_{\mathcal{P}} = (V, E)$, where the node set $V = \{v_1, v_2, ..., v_N\}$ represents the path element in the icon, and $E = \{e_1, e_2, ..., e_{N(N-1)}\}$ represents the set of edges that connecting path elements. Our icon graphs are fully-connected and directed.

4.1.1. Node Features

To extract the feature of a single path elements, we employ the dual-stream GNN proposed in Jiang *et al.* [JLS*21] (Figure 2). They used the dual-stream GNN to extract features and took the relationship between each Bézier curve, such as distance and relative angle, as the model input. The feature vector extracted using [JLS*21] is a 2304-D vector. For more implementation details, please refer to their original paper.

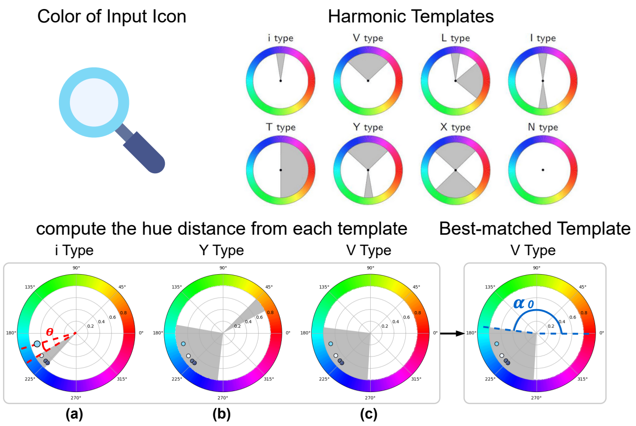

Figure 3: *(a) θ is the hue distance between the color without the sector and the sector boundary. (b) Though all colors fall within the sector, there is a region with no color. In contrast, (c) includes all colors, and the sector is more restrictive. We choose the scheme with the strictest conditions as the best matching one, and record its starting angle as α_0.*

4.1.2. Edge Features

To capture the relative color similarities between path elements, we take the bounding box of each path element, where L and R represent the top-left and bottom-right coordinates of the bounding box, respectively. We capture the following spatial relationship between two path elements and form a 6×1 vector:

$$e_{ij} = [\Delta x_L, \Delta y_L, \Delta x_R, \Delta y_R, \theta_L, \theta_R], \tag{1}$$

where $\Delta x_L = x_L^j - x_L^i$, $\Delta y_L = y_L^j - y_L^i$, $\theta_L = \mathrm{atan2}(\frac{\Delta y_L}{\Delta x_L})$ and $\theta_R = \mathrm{atan2}(\frac{\Delta y_R}{\Delta x_R})$. The edge attribute vectors encode the translation between the two path elements and their relative orientation.

4.1.3. Harmony Template

To preserve the harmonic properties of an input icon, we follow the approach proposed in [COSG*06] by identifying the best matched harmonic template from the seven templates proposed by [Mat95] and [TMI02]. We exclude "type N" because we do not consider black-and-white icons in our work. Each harmonic template T_m and its associated orientation α together define a *harmonic scheme*, denoted as (m, α). We use the following function to measure the harmony of an icon with respect to a specific scheme (m, α):

$$F(\mathcal{P}, (m, \alpha)) = \sum_{p \in \mathcal{P}} \|H(p) - E_{T_m(\alpha)}(p)\| \cdot S(p), \tag{2}$$

where H and S represent the hue and the saturation channels of a path p, respectively. $E_{T_m(\alpha)}$ denotes the sector border hue of template T_m with orientation α that is closest to the hue of path p. The hue distance $\|\cdot\|$ refers to the arc-length distance on the hue wheel, measured in radians. Any hue that falls within the sectors of T_m is considered to have zero distance from the template. By minimizing Equation 2, we determine the best harmonic scheme $(m_{\mathcal{P}}, \alpha_{\mathcal{P}})$ for the input icon $\mathcal{P}$ (Figure 3). For detailed explanation of the optimizing process, please refer to [COSG*06]. Finally, we record $(m_{\mathcal{P}}, \alpha_{\mathcal{P}})$ as the ground truth harmonic properties we intend to preserve.

4.2. Network Architecture

As shown in Figure 2, the colorization template prediction network consists of three parts: (1) node feature extractor, (2) edge feature extractor, and (3) MLP classifier. For the node feature extractor, we use the dual-stream GNN proposed by [JLS*21]. Next, we use two GNN layers to extract the icon graph features, followed by an MLP layer that predicts the probabilities of assigning luminance values and chroma values to each path element.

4.3. Loss Function

Our overall loss function consists of four loss functions:

$$\begin{aligned}\mathcal{L} = {} & w_{\text{unary}} L_{\text{unary}} + w_{\text{pairwise}} L_{\text{pairwise}} \\ & + w_{\text{harmony}} L_{\text{harmony}} + w_{\text{diversity}} L_{\text{diversity}}.\end{aligned} \tag{3}$$

Inspired by previous image segmentation and colorization methods that utilize MRF-based formulations [BJ01, LLW04], we also include a unary term and a pairwise term in our overall loss function.

Unary loss. The goal of this unary term is to encourage the color of each path to match the color in the training data. We use the cross-entropy loss to evaluate each path's predicted color.

Pairwise loss. The goal of the pairwise term is to preserve the color similarity between each neighboring paths in the input icon. We measure the similarity between classes using an exponential function. Specifically, the color similarity between the k-th path pair (i, j) in the input icon is defined as:

$$s_k = \exp(-\alpha \times D_k), \tag{4}$$

where D_k is the Euclidean distance between the classes of path i and path j. The color similarity between the predicted colors of a path pair (i, j) is defined as:

$$\hat{s_k}^{ab} = \frac{1}{2}\left(\frac{\mathbf{\hat{y}_i}^{\mathbf{ab}} \cdot \mathbf{\hat{y}_j}^{\mathbf{ab}}}{\|\mathbf{\hat{y}_i}^{\mathbf{ab}}\|\|\mathbf{\hat{y}_j}^{\mathbf{ab}}\|} + 1\right), \quad \hat{s_k}^{L} = \frac{1}{2}\left(\frac{\mathbf{\hat{y}_i}^{\mathbf{L}} \cdot \mathbf{\hat{y}_j}^{\mathbf{L}}}{\|\mathbf{\hat{y}_i}^{\mathbf{L}}\|\|\mathbf{\hat{y}_j}^{\mathbf{L}}\|} + 1\right) \tag{5}$$

where $\hat{y_i}^{ab}$ and $\hat{y_i}^{L}$ denote the predicted probabilities of ab and L values for path i, respectively. Overall, we measure the distance between all pairs of paths of both L and ab as our loss function:

$$L_{\text{pairwise}} = \sum_{k=1}^{E} [(s_k^{ab} - \hat{s_k}^{ab})^2 + (s_k^L - \hat{s_k}^{L})^2], \tag{6}$$

where E is the number of path pairs.

Color Harmony loss. The goal of the harmony term is to ensure that the predicted color combinations of the icon $\mathcal{P}$ maintain its harmonic properties as defined by the harmonic scheme $(m_{\mathcal{P}}, \alpha_{\mathcal{P}})$. To achieve this, we calculate the hue distance between the predicted color combinations and the best-matched harmonic scheme we identified in the preprocessing stage using:

$$L_{\text{harmony}} = F(\mathcal{P}, (m_{\mathcal{P}}, \alpha_{\mathcal{P}})), \tag{7}$$

where the hue value is derived from the predicted Lab values: $\hat{L}$, $\hat{a}$, and $\hat{b}$. Since the accumulated angular distance can be relatively large compared to other terms, we use the natural logarithm to balance its influence and control strength.

Diversity loss. Given that the paths of the input icon often consist of basic geometric primitives with distinct semantic meanings, it is crucial to ensure that our model does not always assign identical colors for paths that have similar shapes. To encourage diverse color assignments, we use the negative mean of the predicted standard deviation for a and b, which is a common statistical method for measuring diversity:

$$\sigma_a = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{a}_i - \bar{a})^2}, \quad \sigma_b = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{b}_i - \bar{b})^2}, \tag{8}$$

$$L_{\text{diversity}} = -\frac{1}{2}(\sigma_a + \sigma_b), \tag{9}$$

where $\bar{a}$ and $\bar{b}$ represent the mean values of a and b across paths in an icon. It is important to note that if the weight of the this term ($w_{\text{diversity}}$) is too large, the model may lean towards extreme color values. This could result in an overemphasis on colors such as green, red, blue, and yellow, as these combinations maximize the standard deviation.

5. Palette-based Color Transfer

The goal of this step is to obtain various colorization candidates of $\mathcal{P}_{\text{user}}$ using the desired five-color color palette $\mathcal{C}$ for designers to choose from. Specifically, we will generate colorization candidates using the inferred colorization template $\mathcal{L}_{\mathcal{P}_{\text{user}}}$ on the user-provided SVG icon $\mathcal{P}_{\text{user}}$. We denote each colorization candidate as $\mathcal{I} = \{\mathbf{I}_0, \mathbf{I}_1, ..., \mathbf{I}_n\}$, where $\mathbf{I}_i = (L_i, a_i, b_i)$ represent the Lab colors of path i. And we also use $\mathbf{I}_i^L$, $\mathbf{I}_i^a$, and $\mathbf{I}_i^b$ to represent the Lab components of I_i, respectively.

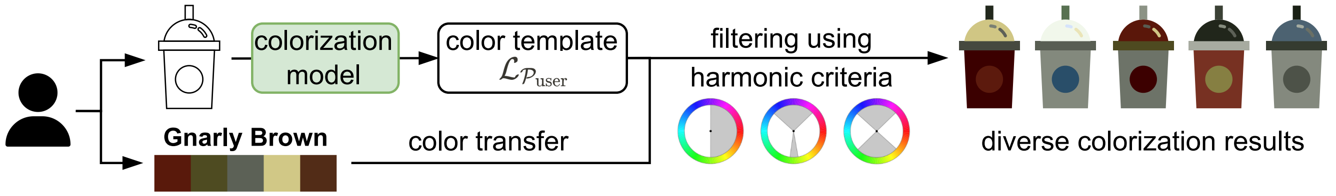

Figure 4: *Given a user-provided SVG icon contour and a five-color palette, our colorization model generates a colorization template, denoted as $\mathcal{L}_{\mathcal{P}_{user}}$. To explore different colorization possibilities, we sample various colorization candidates from $\mathcal{L}_{\mathcal{P}_{user}}$. Next, we perform palette-based color transfer to each colorization candidates. We further perform an filtering process to exclude candidates that do not exhibit harmonious color combinations. Finally, we selected K colorization results that effectively capture the diversity of the colorization space.*

5.1. Generate Colorization Candidates

To generate a variety of colorization candidates for $\mathcal{P}_{\text{user}}$, we sample predicted *Lab* values from the $\mathcal{L}_{\mathcal{P}_{\text{user}}}$. For each path i, we sort the likelihoods l_i of possible *Lab* values and select a subset of *Lab* values based on a pre-defined threshold. This yields a set of sampled *Lab* values denoted as $\psi_i = \{(Lab)_i^0, (Lab)_i^1, ..., (Lab)_i^{k_i}\}$, where k_i is the number of sampled *Lab* values for path i. We repeat this sampling process for all paths in $\mathcal{P}_{\text{user}}$, resulting in a collection of candidate *Lab* values for each path: $\Psi = \{\psi_0, \psi_1, ..., \psi_n\}$. Finally, we generate the final colorization candidates by enumerating all possible combinations from these candidate *Lab* values, denoted as $\Omega = \psi_0 \times \psi_1 \times ... \times \psi_n$.

5.2. Color Transfer

For each colorization candidate in Ω, we proceed by selecting the closest color from the color palette $\mathcal{C}$ for each path ($\mathbf{C}_i$) based on the hue distance metric. We then optimize the final colorization results $\hat{\mathcal{I}}$ by minimizing the following objective function:

$$\arg\min_{\mathcal{I}}\Big(\sum_{i=1}^{n} \|\mathbf{C}_i - \mathbf{I}_i\|_2 + \sum_{i=1}^{n}\sum_{j=1}^{n}\Big(\frac{\lambda}{(\mathbf{I}_i^L - \mathbf{I}_j^L + 1) \times (\|\mathbf{I}_i - \mathbf{I}_j\|_2 + 1)}\Big)\Big) \quad (10)$$

where n is the number of paths, and we set $\lambda = 1000$. The first term penalizes the deviation of the current color values from the corresponding palette colors. The second term is designed to enhance the distinction between assigned colors for individual paths by increasing the disparities in their luminance values. This term is motivated by the objective of ensuring that paths with the same corresponding palette colors yield colorization results that maintain harmony and recognition [OL06]. Additionally, we aim to preserve the relative luminance relationships between paths during color transfer process [RAGS01, TJT05, CFL*15]. For example, if path i is predicted to be lighter than path j, it is crucial that after recoloring, path i remains lighter than path j.

5.3. Candidate Filtering and Selection

After color transfer, we observed that not all colorization align with existing harmonic templates. Therefore, we exclude colorization results that do not match any of the existing harmonic templates. In order to enhance designers' workflow, it is crucial to avoid overwhelming them with an excessive number of colorization results, as it may lead to confusion. Based on our interview with designers, we discovered that owners often struggle to express their specific needs or preference accurately. Additionally, designers often face challenges in meeting the owners' requirement in a single attempt, often necessitating multiple iterations. To mitigate the communication burden between designers and owners, we propose offering a range of colorization results that designers can directly present to owners for selection. To determine the final colorization results, we use the following selection process. First, we select the colorization result from the set of candidates in Ω that has the the highest likelihoods for every paths. Next, we calculate the L2 distances between the selected colorization result and the remaining candidates. The candidate with the largest distance is then chosen as the subsequent colorization result. We iterate this process K times, allowing designers to specify the desired number of K. This approach enable us to present a diverse set of colorization results.

6. Experiments

6.1. Datasets

Vector icons. We collected 13,132 icons from Flaticon [†], a public website of icon creations by designers. All icons are in SVG format with flat colors and no borders. To prepare the dataset, we extracted the contour of each icon and set the stroke color as black with a width to six. For training, validation, and testing, we splitted the data in an 8:1:1 ratio.

Palettes. We use the Kuler dataset provided by O'Donovan *et al.* [OAH11] to demonstrate the colorization results. Each palette consists of a theme name and five distinct RGB colors.

6.2. Implementation Details

Our model was trained using a GeForce RTX 1080 Ti GPU on Ubuntu 20.04.1. We employed PyTorch framework with batch size 8. The training lasted for 50 epochs, took approximately 1.5 hours. We use Adam optimizer and set the learning rate as 2.5×10^{-4}. We set the weights of unary, pairwise, diversity, and harmony terms are set to 0.2, 0.8, 0.01, and 0.01, respectively.

† https://www.flaticon.com/

Figure 5: *Given an icon set, designers can use our method to efficiently evaluate the colorization results using different palettes.*

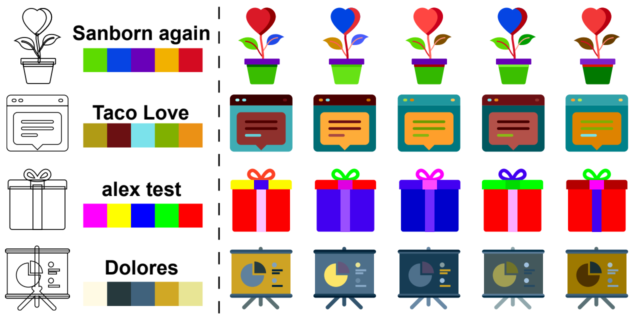

Figure 6: *Given an icon contour and a desired palette, our method generate numbers of colorization results. This enables designers to explore and evaluate various colorization possibilities for the given icon and palette pair.*

6.3. Performance

In the test phase, our model predicts the probability of each color for each path. It takes about 28 seconds for 1313 icons and an average of about 0.02 seconds per icon. The color transfer process for two icons, one with 5 path elements and the other with 12 path elements:

- For colorizing a single icon (solving Equation 10), the processing time was 1.2 and 3 seconds.
- Filtering a thousand colorization results using harmonic templates took 27 seconds and 1.5 minutes.
- Providing k different colorization results to users took 0.2 and 0.3 seconds.

6.4. Results

In Figure 5 and Figure 6, we present the colorization results generated by our method. Our method can be used in the scenario where designers aim to colorize an icon set comprises multiple icons using multiple color palettes. We present the coloring results in Figure 5 to designers, who have provided valuable feedback on our approach. They mentioned that our method allows them to quickly and easily identify the optimal palette for an icon set, thus significantly reducing their workload in searching suitable color palettes.

As mentioned in Section 5.3, communication between designers and owners incurs significant costs. Multiple transmission and modification rounds are often required to capture the owners' demands effectively. In response to this situation, we propose to provide a series of colorization results that designers can directly present to owners for selection. Figure 6 shows examples of diverse colorization candidates. Designers can adjust the number of candidates for different situations. Owners can select their preferred icons from these options to facilitate the elucidation of requirements. Designers can subsequently utilize the chosen icons directly for color modifications.

6.5. Evaluation

We compare our colorization results to several iconic line art colorization works, including Comic [FHOO17], MUNIT [HLBK18], AdvIcon [SLWW19], SCFT [LKL*20], SGA [LGK*22], and IconFlow [LLW22]. For each method, we use the implementations provided by the authors. Given that previous studies may not directly focus on icon colorization, we finetune each method using the pixel icon dataset provided by Sun *et al.* [SLWW19]. In addition, previous works require another image as a color reference, while we use the color palette directly. We select some icons as color references for comparison baselines to present the coloring results. Then, we use k-means to take five representative colors from these reference icons as our palette input. Figure 7 shows all coloring results. Regarding the quantitatively assessing color harmony, a notable challenge emerges due to the inherent inconsistency of color distribution between the colorization results of our method (sparse set of colors) and traditional "pixel-based" methods. Although we try to compute the harmony distance using selected representative colors (following [CFL*15]), we discovered that this approach fails

Figure 7: *We compare the colorization results obtained by our vector icon colorization methods that operate on pixel icons.*

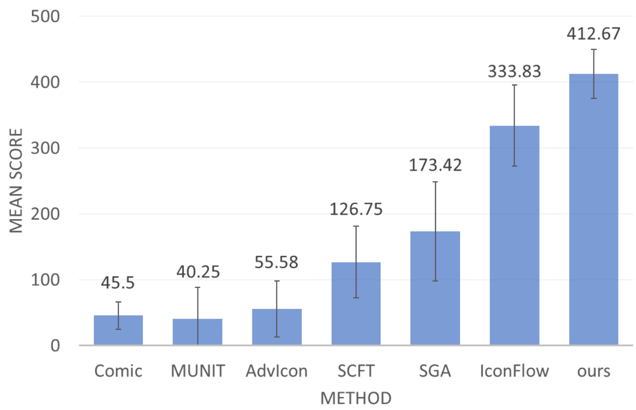

Figure 8: *We visualize the mean and the standard deviation of the user study results. In comparison to other baseline methods, our method obtain the highest score.*

to fully capture the perceptual image harmony, mainly due to the presence of noisy colors in the colorization results.

6.6. User Study

We conducted an anonymous questionnaire to evaluate the colorization results of our method compared to several baseline methods. The questionnaire consists of 12 questions using the 12 examples shown in Figure 7. Each question presents an icon contour and a palette, along with seven options: our coloring results and the results of six baseline methods. Prior to the questionnaire, we explained that the colorization results were drawn based on the given icon contour and the palette without any additional hints or reference standards for good colorization. To prevent visual fatigue, we asked participants to choose the three best colorization results and rank them from good to bad, rather than ranking all seven images at once. Our scoring method award three points for a first place ranking, two points for second place, and one point for third place. Icons that were not selected by the participants were not included in the score calculation.

In total, we recruited 198 participants, including 68 males, 125 females, and five did not declare their gender. The participants' ages ranged from 15 to 34, with two respondents are over the age of 40. The results of the questionnaire, shown in Figure 8, revealed our method obtained the highest mean score. These observations indicate a general preference for our colorization results among the majority of participants, highlighting the effectiveness of our approach in generating visually appealing colorizations.

6.6.1. Discussion

In contrast to our method that directly operate on vector icons, the baseline methods require rasterizing the input vector icon into a pixel icon before colorization. As a result, the colorization results obtained by baseline methods are in pixel format. Therefore, designers need to manually refill the resulting colors from the pixel images to the vector icons for further usage. After the user study, we collected the criteria for judging the colorization results from the participants. Some participants highlighted their consideration of the colorization method's ability to preserve the structure of the original icon contour. This include aspects such as whether the contour is complete, whether the hole is preserved, and whether colors of the overlapping paths are blended. Our method preserve these aspects better compared to baseline methods operate on pixel icons. Moreover, some participants expressed their consideration of the

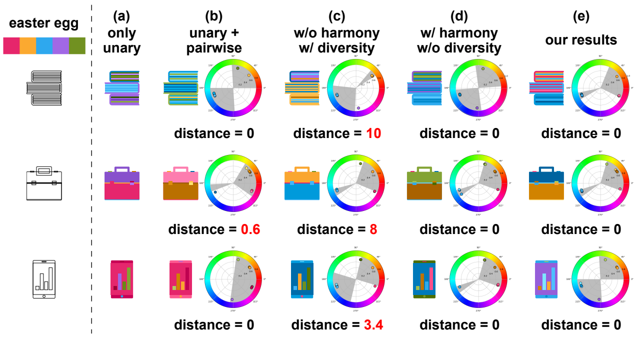

Figure 9: *Ablation studies. (a) Using only the unary term may result in indistinguishable shapes. (b) Consider relationships between paths using pairwise terms. (c) Enhancing color variations by introducing diversity term. (d) Improving the harmony of the resultant color combinations using harmony term. (e) Colorization results using our overall loss function. From (b) to (e), we compute the hue distance between each example's colors and its best matched-template and plot the results on the hue wheel.*

harmoniousness of color combinations in the resulting colorizations. Although color harmony is subjective, our method endeavors to maintain it and achieve the closest match to human perception of color compared to baseline methods.

6.7. Ablation Study

We conducted ablation studies to analyze the contributions of different losses in our vector icon colorization method, including pairwise, harmony, and diversity losses (Figure 9). When only the unary term is used, there are challenges in distinguishing the contours of path due to the similarity in predicted luminance and chroma values. By introducing the pairwise loss, our method is able to consider the relative relationship between paths to differentiate the shapes' color clearly (Figure 9(b)). The harmony loss helps align the colorization results with human-preferred color perception, resulting in more visually appealing outcomes (Figure 9(d)). We observe that when only unary and pairwise losses are used to train the colorization model, the icons tend to be biased towards a specific color. For instance, in Figure 9(b), the book is predominantly blue, and the mobile phone appears reddish overall. With the diversity loss, the colorization model can generate various colors, enhancing the overall diversity of the output color combinations. By integrating unary, pairwise, harmony, and diversity losses for training our colorization model, we ensure that the resulting color templates can be used to generate a range of visual appealing colorization results with the reference palette.

6.7.1. Harmony Evaluation

To validate the effectiveness of introducing the harmony loss in enhancing color combinations, we calculate the hue distance between each color combination in the examples and its best-matched template from Figure 9(b) to Figure 9(e). When we only include the unary and pairwise terms in the loss function, only one example that yields a non-zero hue distance, indicating it does not meet the

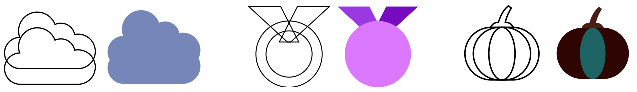

Figure 10: *Due to the color discretization during training, our method is limited in capturing subtle color differences between paths. As a result, when similar shapes overlap, they often become indistinguishable as they are predicted to have the same color. For example, this can be observed in cases such as two overlapping clouds (left), two circles on the medal (middle), and similar shapes on the pumpkin (right).*

harmonious criterion. However, due to the limited color variations, the inclusion of the diversity term becomes imperative. Nonetheless, this inclusion results in an amplified hue distance as illustrated in Figure 9(c) (mean hue distance is 7.1°). After introducing harmony loss, all resultant color combinations (Figure 9(d)) meet the harmonious criterion (mean hue distance is 0), while simultaneously maintaining color diversity.

7. Limitations and Future Work

Distinction of similar shapes. As described in Section 4, we discretized the *Lab* colors into a fixed number of classes and treat the colorization problem as a classification problem. However, due to the discretization, the pairwise loss cannot preserve the color differences within each class. As a result, our model may assign the same color to paths with similar shapes, leading to difficulties in distinguish these shapes when they overlap with each other. In the future, it is possible to explicitly encourage different color assignments by encoding the overlap relationship in the loss functions.

Limited Color Model. Currently, our method supports constant color model, where a single color is assigned to each path in the vector icon. In the future, we plan to extend our colorization method to also support the gradient color model. Furthermore, during our interview with designers, we received feedback suggesting that style transfer on vector icon would be an exciting direction to explore. Designers mentioned that the owner often provide reference images as style sources, rather than providing concrete style descriptions.

Text-based Vector Icon Colorization. During our interview with designers, they mentioned that owners often express their requirements for colorization in abstract text descriptions rather than providing specific reference color palettes. This observation has led us believe that a promising future direction to explore is the vector icon colorization using text inputs.

Perceptual Similarity Between Colors. Currently, we use a cross-entropy loss to model the colorization problem because it makes the colorization trackable, as demonstrated in [ZIE16]. We believe a more refined formulation considering the perceptual similarity between different colors is a fruitful future work.

Match color palettes and harmonic templates. We currently select the colorization results by sequentially performing palette transfer and harmonic filtering rather than incorporating it due to occasional discrepancies observed in the colorization results, where the inclusion of the harmonic template matching term caused excessive deviation from both the predicted color and the color palette. We believe there are still many possible ways to achieve different benefits. For example, adjusting the color to match the palette and harmonic template simultaneously is a potential alternative to transfer colors.

8. Conclusion

We present a palette-based colorization method for colorizing vector icons without the need of rasterization. Our method consists of two stages. First, we introduced a colorization model that predicts a color template based on the input icon contour. We simultaneously combine vector icons' local and group features to strengthen the feature descriptions. To encourage colorization results that align with the human perception of color harmony,we design a novel loss function comprises a MRF-based colorization loss and a color harmony loss. In the second stage, we perform color transfer from a desired palette to the color template and generate a diverse range of colorization results. Our method provide valuable support to designers in two ways. First, designers can explore various colorziation results for a single icon using different color palettes. Second, they can efficiently evaluate the suitability of a color palette for a set of icons. We conducted a user study to evaluate the colorization results generated by our method and several baseline methods. The results clearly indicated that the results generated by our method are the most preferred among all the baseline methods. More importantly, our proposed method seamlessly integrates into designers' workflow, allowing them to modify colorization results directly for subsequent icon design tasks, thereby enhancing the ease and efficiency of vector icon colorization.

9. Acknowledgement

We thank the anonymous reviewers for their valuable feedback and Fu-Yin Cherng for insightful discussion. This work was supported in part by JSPS Grant-in-Aid JP23K16921, and National Science and Technology Council, under Grant, NSTC111-2634-F-002-022, 111-2221-E-002-145-MY3, 111-3111-E-002-002, and 112-2218-E-002-029, and National Taiwan University (NTU112L900902).

References

[BJ01] Boykov Y. Y., Jolly M.-P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001* (2001), vol. 1, IEEE, pp. 105–112. 5

[BPCD11] Bychkovsky V., Paris S., Chan E., Durand F.: Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011* (2011), IEEE, pp. 97–104. 3

[BPD06] Bae S., Paris S., Durand F.: Two-scale tone management for photographic look. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 637–645. 3

[CBJ19] Collomosse J., Bui T., Jin H.: Livesketch: Query perturbations for guided sketch-based visual search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 2879–2887. 3

[CDAT20] Carlier A., Danelljan M., Alahi A., Timofte R.: Deepsvg: A hierarchical generative network for vector graphics animation. *Advances in Neural Information Processing Systems 33* (2020), 16351–16361. 3

[CFL*15] Chang H., Fried O., Liu Y., DiVerdi S., Finkelstein A.: Palette-based photo recoloring. *ACM Trans. Graph. 34*, 4 (2015), 139–1. 3, 6, 7

[CMW*18] Ci Y., Ma X., Wang Z., Li H., Luo Z.: User-guided deep anime line art colorization with conditional adversarial networks. In *Proceedings of the 26th ACM international conference on Multimedia* (2018), pp. 1536–1544. 3

[Coc14] Cochrane S.: The munsell color system: A scientific compromise from the world of art. *Studies in History and Philosophy of Science Part A 47* (2014), 26–41. 2

[COSG*06] Cohen-Or D., Sorkine O., Gal R., Leyvand T., Xu Y.-Q.: Color harmonization. In *ACM SIGGRAPH 2006 Papers*. 2006, pp. 624–630. 2, 3, 5

[CSMS10] Csurka G., Skaff S., Marchesotti L., Saunders C.: Learning moods and emotions from color combinations. In *Proceedings of the seventh Indian conference on computer vision, graphics and image processing* (2010), pp. 298–305. 3

[CYMLYC17] Cho J., Yun S., Mu Lee K., Young Choi J.: Palettenet: Image recolorization with given color palette. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (2017), pp. 62–70. 3

[FHOO17] Furusawa C., Hiroshiba K., Ogaki K., Odagiri Y.: Comicolorization: semi-automatic manga colorization. In *SIGGRAPH Asia 2017 Technical Briefs*. 2017, pp. 1–4. 7

[HA17] Hensman P., Aizawa K.: cgan-based manga colorization using a single training image. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* (2017), vol. 3, IEEE, pp. 72–77. 2

[HE17] Ha D., Eck D.: A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477* (2017). 3

[HLBK18] Huang X., Liu M.-Y., Belongie S., Kautz J.: Multimodal unsupervised image-to-image translation. In *ECCV* (2018). 7

[HSGL11] HaCohen Y., Shechtman E., Goldman D. B., Lischinski D.: Non-rigid dense correspondence with applications for image enhancement. *ACM transactions on graphics (TOG) 30*, 4 (2011), 1–10. 3

[HZZ20] Han Q.-R., Zhu W.-Z., Zhu Q.: Icon colorization based on triple conditional generative adversarial networks. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)* (2020), IEEE, pp. 391–394. 3

[ISSI16] Iizuka S., Simo-Serra E., Ishikawa H.: 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), 1–11. 2

[Itt70] Itten J.: *The elements of color*, vol. 4. John Wiley & Sons, 1970. 2

[IZZE17] Isola P., Zhu J.-Y., Zhou T., Efros A. A.: Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 1125–1134. 2

[JLS*21] Jiang X., Liu L., Shan C., Shen Y., Dong X., Li D.: Recognizing vector graphics without rasterization. *NeurIPS 34* (2021), 24569–24580. 3, 4, 5

[JO48] Jacobson E., Ostwald W.: *Color harmony manual*. Container Corporation of America Chicago, 1948. 2

[KJPY19] Kim H., Jhoo H. Y., Park E., Yoo S.: Tag2pix: Line art colorization using text tag with secat and changing loss. In *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 9056–9065. 3

[KM16] Kita N., Miyata K.: Aesthetic rating and color suggestion for color palettes. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 127–136. 2

[KYKL14] Kim H.-R., Yoo M.-J., Kang H., Lee I.-K.: Perceptually-based color assignment. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 309–318. 2

[LFUS06] Lischinski D., Farbman Z., Uyttendaele M., Szeliski R.: Interactive local adjustment of tonal values. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 646–653. 3

[LGK*22] Li Z., Geng Z., Kang Z., Chen W., Yang Y.: Eliminating gradient conflict in reference-based line-art colorization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII* (2022), Springer, pp. 579–596. 2, 7

[LHES19] Lopes R. G., Ha D., Eck D., Shlens J.: A learned representation for scalable vector graphics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 7930–7939. 3

[LKL*20] Lee J., Kim E., Lee Y., Kim D., Chang J., Choo J.: 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* (2020), pp. 5801–5810. 2, 7

[LLW04] Levin A., Lischinski D., Weiss Y.: Colorization using optimization. In *ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), SIGGRAPH '04, Association for Computing Machinery, p. 689–694. 5

[LLW22] Li Y.-k., Lien Y.-H., Wang Y.-S.: Style-structure disentangled features and normalizing flows for diverse icon colorization. In *CVPR 2022* (2022). 1, 2, 3, 7

[LQLW17] Liu Y., Qin Z., Luo Z., Wang H.: Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks. *arXiv preprint arXiv:1705.01908* (2017). 2

[LRFH13] Lin S., Ritchie D., Fisher M., Hanrahan P.: Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 1–12. 2

[LZNH15] Li X., Zhao H., Nie G., Huang H.: Image recoloring using geodesic distance based color harmonization. *Computational Visual Media 1* (2015), 143–155. 3

[LZZ*18] Li L., Zou C., Zheng Y., Su Q., Fu H., Tai C.-L.: Sketch-r2cnn: An attentive network for vector sketch recognition. *arXiv preprint arXiv:1811.08170* (2018). 3

[Mat95] Matsuda Y.: Color design. *Asakura Shoten 2*, 4 (1995), 10. 2, 5

[MS44] Moon P., Spencer D. E.: Geometric formulation of classical color harmony. *JOSA 34*, 1 (1944), 46–59. 2

[OAH11] O'Donovan P., Agarwala A., Hertzmann A.: Color compatibility from large datasets. In *ACM SIGGRAPH 2011 papers*. 2011, pp. 1–12. 2, 6

[OCLM11] Ou L.-C., Chong P., Luo M. R., Minchew C.: Additivity of colour harmony. *Color Research & Application 36*, 5 (2011), 355–372. 2

[OL06] Ou L.-C., Luo M. R.: A colour harmony model for two-colour combinations. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur 31*, 3 (2006), 191–204. 2, 6

[RAGS01] Reinhard E., Adhikhmin M., Gooch B., Shirley P.: Color transfer between images. *IEEE Computer graphics and applications 21*, 5 (2001), 34–41. 3, 6

[RBCP20] Ribeiro L. S. F., Bui T., Collomosse J., Ponti M.: Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 14153–14162. 3

[RGLM21] Reddy P., Gharbi M., Lukac M., Mitra N. J.: Im2vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 7342–7351. 3

[SC21] Shen I.-C., Chen B.-Y.: Clipgen: A deep generative model for clipart vectorization and synthesis. *IEEE Transactions on Visualization and Computer Graphics 28*, 12 (2021), 4211–4224. 3

[SLWW19] Sun T.-H., Lai C.-H., Wong S.-K., Wang Y.-S.: Adversarial colorization of icons based on contour and color conditions. In *Proceedings of the 27th ACM International Conference on Multimedia* (2019), pp. 683–691. 1, 3, 7

[TJT05] Tai Y.-W., Jia J., Tang C.-K.: Local color transfer via probabilistic segmentation by expectation-maximization. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 1, IEEE, pp. 747–754. 3, 6

[TMI02] Tokumaru M., Muranaka N., Imanishi S.: Color design support system considering color harmony. In *FUZZ-IEEE'02* (2002), vol. 1, IEEE, pp. 378–383. 2, 5

[WJC13] Wang X., Jia J., Cai L.: Affective image adjustment with a single word. *The Visual Computer 29* (2013), 1121–1133. 3

[WL21] Wang Y., Lian Z.: Deepvecfont: Synthesizing high-quality vector fonts via dual-modality learning. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–15. 3

[WYW*10] Wang B., Yu Y., Wong T.-T., Chen C., Xu Y.-Q.: Data-driven image color theme enhancement. *ACM Transactions on Graphics (TOG) 29*, 6 (2010), 1–10. 3

[WYX11] Wang B., Yu Y., Xu Y.-Q.: Example-based image color and tone style enhancement. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 1–12. 3

[XHY*18] Xu P., Huang Y., Yuan T., Pang K., Song Y.-Z., Xiang T., Hospedales T. M., Ma Z., Guo J.: Sketchmate: Deep hashing for million-scale human sketch retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 8090–8098. 3

[YPCL13] Yoo J.-D., Park M.-K., Cho J.-H., Lee K. H.: Local color transfer between images using dominant colors. *Journal of Electronic Imaging 22*, 3 (2013), 033003–033003. 3

[ZIE16] Zhang R., Isola P., Efros A. A.: Colorful image colorization. In *ECCV 2016* (2016), Springer, pp. 649–666. 2, 4, 9

[ZJLL17] Zhang L., Ji Y., Lin X., Liu C.: Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In *2017 4th IAPR Asian conference on pattern recognition (ACPR)* (2017), IEEE, pp. 506–511. 2

[ZLW*18] Zhang L., Li C., Wong T.-T., Ji Y., Liu C.: Two-stage sketch colorization. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–14. 2

[ZSQ*21] Zou Z., Shi T., Qiu S., Yuan Y., Shi Z.: Stylized neural painting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 15689–15698. 3

[ZXST17] Zhang Q., Xiao C., Sun H., Tang F.: Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing 26*, 4 (2017), 1952–1964. 3

[ZYZ*17] Zhang X.-Y., Yin F., Zhang Y.-M., Liu C.-L., Bengio Y.: Drawing and recognizing chinese characters with recurrent neural network. *IEEE transactions on pattern analysis and machine intelligence 40*, 4 (2017), 849–862. 3

[ZZI*17] Zhang R., Zhu J.-Y., Isola P., Geng X., Lin A. S., Yu T., Efros A. A.: Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999* (2017). 2