Color Recommendation for Vector Graphic Documents based on Multi-Palette Representation

Qianru Qiu
CyberAgent
Shibuya, Tokyo, Japan
qiu_qianru@cyberagent.co.jp

Xueting Wang
CyberAgent
Shibuya, Tokyo, Japan
wang_xueting@cyberagent.co.jp

Mayu Otani
CyberAgent
Shibuya, Tokyo, Japan
otani_mayu@cyberagent.co.jp

Yuki Iwazaki
CyberAgent
Shibuya, Tokyo, Japan
iwazaki_yuki@cyberagent.co.jp

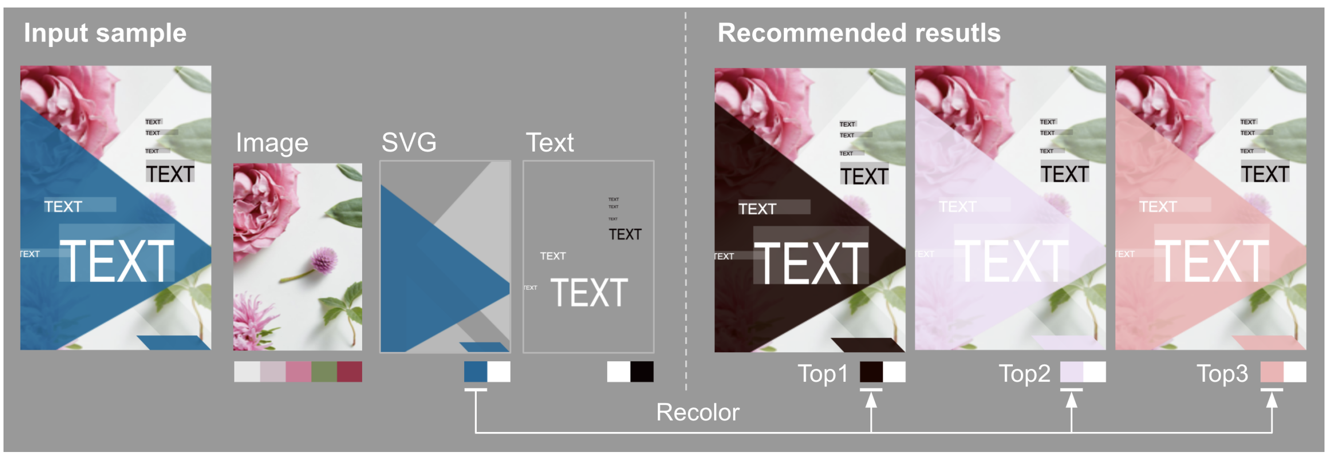

Figure 1: Color recommendation for vector graphic documents. On the left are the design sample and its visual elements (e.g. Image, SVG and Text) with extracted palettes. On the right are the recommended top3 results recoloring one SVG color. The elements in the input sample are from the Crello dataset.

Abstract

Vector graphic documents present multiple visual elements, such as images, shapes, and texts. Choosing appropriate colors for multiple visual elements is challenging but crucial for amateurs and professional designers. Instead of creating a single color palette for all elements, we extract multiple color palettes from each visual element of a graphic document, and then combine them into a color sequence. We propose a masked color model for color sequence completion and recommend the specified colors based on the color context in multi-palette with high probability. We train the model and build a color recommendation system on a large-scale dataset of vector graphic documents. The proposed color recommendation method outperformed other state-of-the-art methods by quantitative and qualitative evaluations on color prediction and our color recommendation system received positive feedback from professional designers in an interview study. Our code and trained model are available at `https://github.com/CyberAgentAILab/multipalette`.

1. Introduction

In graphic design, many creative applications provide numerous templates. These design platforms suit creative designers and amateurs such as marketing professionals, bloggers, social media managers, etc. In design workflows, the users choose a template and replace the elements using their own assets. The predesigned templates have coordinated colors in each visual element. When some visual elements are replaced, the color harmony may be destroyed. Selecting appropriate colors may not be easy for amateurs; even designers usually struggle with getting suitable color palettes for vector graphic documents.

A color palette refers to a limited number of colors expressed in refined forms. It is widely used in graphic design owing to its simplicity, intuitiveness, generality, and easy computation [12]. The prior researches on color palette representation proposed to train a regression model [16, 13]. These regression methods extract hundreds of color features manually and learn the weights of each feature. The feature extraction is complex, including palette colors, mean, standard deviation, median, max, min, and max minus min across a single channel in each color space, i.e., RGB, CIELAB, and HSV. The extraction of the hand-crafted features is difficult because they do not comprehensively encode the semantics of colors, and some features might not have significant effects on the downstream tasks. Learning high-quality representation of color remains an open problem. In this study, we simplify the input without hand-crafted features and propose a data-driven deep learning model for color representation.

In recent years, researchers have explored deep learning techniques for color palette generation and color recommendation. Previous researches focus on generating a color palette for a single visual target, such as image colorization [5], shape colorization in statistical graphics [14], shape and text colorization in infographics [22]. However, a vector graphic document is much more complex with multiple visual elements, including images, shapes, and texts. Each visual element has its palette. It is challenging for existing color recommendation tools to recommend colors for a multi-palette design. In this study, we use a color sequence combining multiple palettes of different visual elements and train a masked color model to learn multi-palette representation by color sequence completion.

In summary, our main contributions include:

- A novel masked color model to represent multiple palettes in vector graphic documents.
- An interactive system for color recommendation which recolors visual elements with the recommended colors.
- A series of experimental evaluations covering quantitative experiments and perceptual studies for the recommendation system and the recommended results that validate the proposed methods' effectiveness.

2. Related works

2.1. Color recommendation

There are mainly two scenarios of color recommendation. The first is to suggest a color palette for specified themes or semantic requirements. The second is to expand a color palette based on the given colors. For the first scenario, some websites like Adobe Color [1] and COLOURLovers [2], provide color palette templates classified with various theme names or semantic tags, such as 'natural' and 'environmental'. These palette templates can be used as references to suit semantic requirements. Some researchers suggest color palette templates based on semantic tags and fixed harmonic color selection models to generate text colors based on image colors in magazine cover design [8, 21]. For the second scenario, an early effort by O'Donovan *et al.* [16] proposed a linear regression method and suggested a fifth color for the given four colors. Kita *et al.* [13] used the same regression method to expand a color palette composed of N colors to $N + \alpha$, retraining the original color harmony. These regression models depended on the hand-crafted feature extraction methods. In this work, we suggest compatible colors for the given colors in multi-palette without color feature extraction methods by a deep learning model.

Recently, some researchers have explored deep learning algorithms for color palette recommendations. Yuan *et al.* [22] employed a Variational AutoEncoder with Arbitrary Conditioning (VAEAC) model to generate a color palette for infographic elements. Kim *et al.* [11] trained the color embedding model to predict and recommend other colors that are likely to gather together in the same palette for Mandala Coloring. The color model was similar to fastText [6], an extension of the Word2Vec [15] model. They signified a color as a word, and the palette as a sentence. The model was trained providing a continuous vector representation of colors. The elements in these design objects include shapes or texts, and each element is limited to a single color. In graphic documents, the elements also include photos and illustrations. The color design in vector graphic documents is more complex than a multi-palette design. We extend the idea of applying word embedding to color representation similar to the previous Word2Vec-based work [11]. However, the Word2Vec models cannot account for the relationships between different palettes in the same graphic document. We explore multi-palette representation by contextual embedding based on BERT architecture [9].

2.2. Palette-based image recoloring

Some image recoloring approaches are based on semantic segmentation from the image by deep neural networks [3, 10]. We focus on palette-based models as the color concept of an image is easy to express with a color palette. A color palette captures the principal colors in an image and adjusts the color composition towards the desired color palette. Most approaches for recoloring images involve two steps: extracting a palette from the image and mapping every pixel in the image to the target palette. Many strategies

[7, 23, 4] employ clustering methods to extract palette colors. Several other works [17, 18] use a geometric approach to extract palettes that construct a convex hull in RGB color space. The convex-hull-based palettes may miss important colors that lie within the convex hull. We use the k-means clustering method to recolor image elements in our system.

3. Multi-palette representation

3.1. Datasets

We generate a multi-palette dataset from Crello [19], a large-scale dataset containing design templates for various display formats, such as social media posts, banner ads, blog headers, and printed posters. It offers complete document structure and element attributes including element-specific configuration, such as the element type, position, size, opacity, color, or a raster image. The element types mainly include imageElement, maskElement, coloredBackground, svgElement, and textElement. We classify the elements into three groups, as image element group including imageElement and maskElement, SVG element group including coloredBackground and svgElement, and text element group including textElement. The color data of each element in the Crello dataset only has one color that is relevant for solid background and text placeholder. We generate a multi-palette dataset as Image-SVG-Text palettes that each element group has its own palette as shown in Figure 2. For image and SVG elements, we merge the elements of the same group into a single image and then extract the color palette using the k-means clustering method. We use $k = 5$ that it works well for typical elements. We collect text colors and cluster them into a palette for text elements. Each palette is up to five colors in this work. We get 18,768 / 2,315 / 2,278 valid data as train, validation, and test datasets. All design templates in the figures of this paper are from the Crello test dataset.

3.2. Representation learning with masked color model

We train the color embedding model similar to the word embedding model. In natural language processing, the word embedding model is used to learn distributed representation, where the input is a text corpus and the output is a set of feature vectors representing words. Similarly, in the color embedding model, a color signifies a word, a palette signifies a sentence, and multiple palettes in the same design signify a paragraph.

We adopt the CIELAB color space for the input color corpus, which is more perceptually uniform than other color spaces [12]. The most commonly used color space is the 24-bit RGB model. We convert RGB color data to CIELAB with a range of [0, 255], and assign each color to one of the bins in a $b \times b \times b$ histogram (we use $b = 16$ in this work).

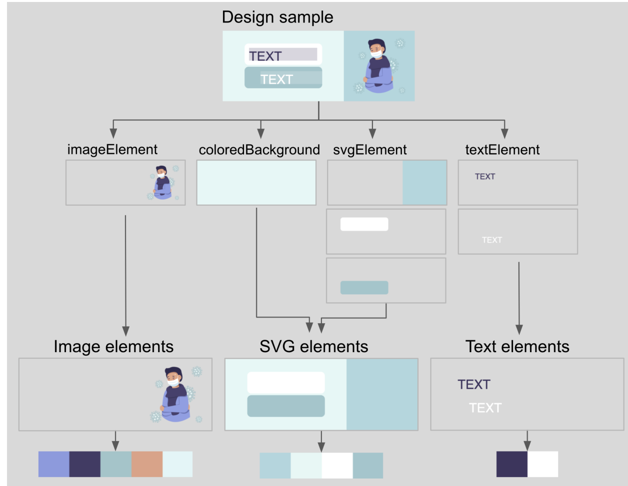

Figure 2: Multi-palette extraction from a design template as Image-SVG-Text palettes. Merge the elements in the same element group into a single image and then extract the color palette.

For example, the color white(255, 255, 255) in RGB color space is labeled as the code $'15\_8\_8'$ in CIELAB color space with 16 bins. There are 796 color codes in the vocabulary of the training dataset. The color codes are converted into vectors and embedded in the space in the learning progress.

We obtain color embeddings by a masked color model based on pre-training BERT architecture [9]. The masked color model in Figure 3 is trained similarly to the masked language model (Masked LM) in BERT. The model receives a fixed length of each palette as input. For a palette that is shorter than this fixed length, we will have to add the token [PAD] to the palette to make up the length. Another artificial token [SEP], is added to the end of a palette. The maximum sequence length is 18. Its input representation is constructed for a given token by summing the corresponding token, segment, and position embeddings. Here, $\{C_{1_1}, \ldots C_{1_5}\}$ is for the image color palette, $\{C_{2_1}, \ldots C_{2_4}\}$ is for the SVG color palette, and $\{C_{3_1}, C_{3_2}\}$ is for the text color palette. The image, SVG, and text palettes, are labeled with segment numbers 1, 2, and 3. The segment embeddings are the palette numbers encoded into a vector. The trained model knows whether a particular color token belongs to a specific palette. Segment embeddings can achieve multi-palette representation.

The BERT-based model architecture is a multi-layer bidirectional Transformer encoder. We use four transformer layers and eight self-attention heads. Increasing the number of transformer layers or attention heads doesn't have a significant effect on the accuracy of model in this work.

The masked color model randomly masks some percent-

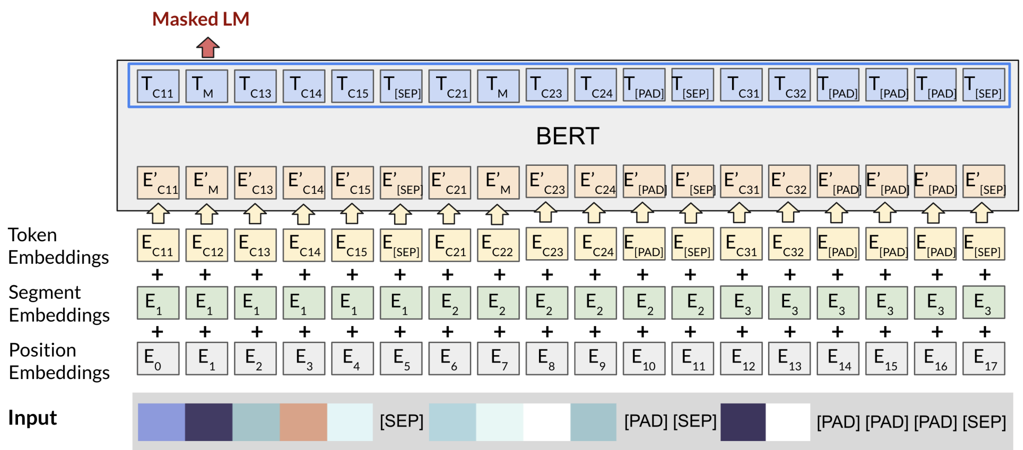

Figure 3: Masked color model for image-SVG-text color sequence.

age of the tokens from the input, and then predicts the masked tokens based on their context. In our experiments, we randomly mask 10% of the tokens in each sequence and replace the chosen token with the [MASK] token 80% of the time. Then, we use the standard cross entropy loss to optimize the pre-training task. We recommend colors in multi-palette by predicting masked colors with high probability.

3.3. Color recommendation system

Users can choose and edit a design template in the existing creative applications for graphic documents. However, when the same visual elements are replaced, users may struggle with coordinating the colors in the design. To reduce the user's work, we propose a system to recommend the specified colors and automatically recolor the elements with the suggested colors.

We create a color recommendation engine with a masked color model and develop an interactive user interface that lets users obtain the coordinated colors for the design. The color recommendation system supports basic selecting, interactive recommendation, and previewing functions shown in Figure 4. The design template is converted to a JSON file as the system input that contains the complete element-specific configuration. The system parses a JSON object and reconstructs the design with separated visual elements. The system allows users to change the image elements and extracts the color palette of each element group as step two in Figure 4. The image palette shown below is extracted from the new image. Users can select the colors for recoloring and then check the design results with the recommended colors. For SVG recoloring, a simple interpolation method changes the original color to the recommended color. We use a palette-based photo recoloring method by k-means clustering [7] in this system for image recoloring.

4. Experimental validation

To evaluate the performance of our proposed approach, we compare it with related work and a baseline model by quantitative and qualitative evaluations. We adapted a Word2Vec-based model which is used in the related work [11]. The input for this model is the color token without segment embeddings. We also trained a BERT-based model without segment embeddings as a baseline to show the effectiveness of the segment embeddings for multi-palette representation.

4.1. Quantitative evaluation

We use 2278 color sequences of our test dataset in the quantitative experiments. We randomly mask a color in the color sequence and evaluate the predicted color's accuracy. We use top N accuracy that the actual color equals any of the N most probable colors predicted by each model. The human eye sometimes cannot fully perceive subtle color differences. In addition to accuracy, we also use visual similarity to measure the recommended colors. For similarity measurement, we calculate the distance between two colors using CIEDE2000 rather than Euclidean distance, which has exhibited good performance in predicting visual similarity between color palettes [20, 12].

Firstly, we train 20 times and get the mean value of accuracy. The results of our model with and without segment and position embeddings are shown in Table 1. It is found that there is no difference between the models with and without position embeddings in the current dataset. Thus we pick up the best models trained by our method without position embeddings for the following comparisons.

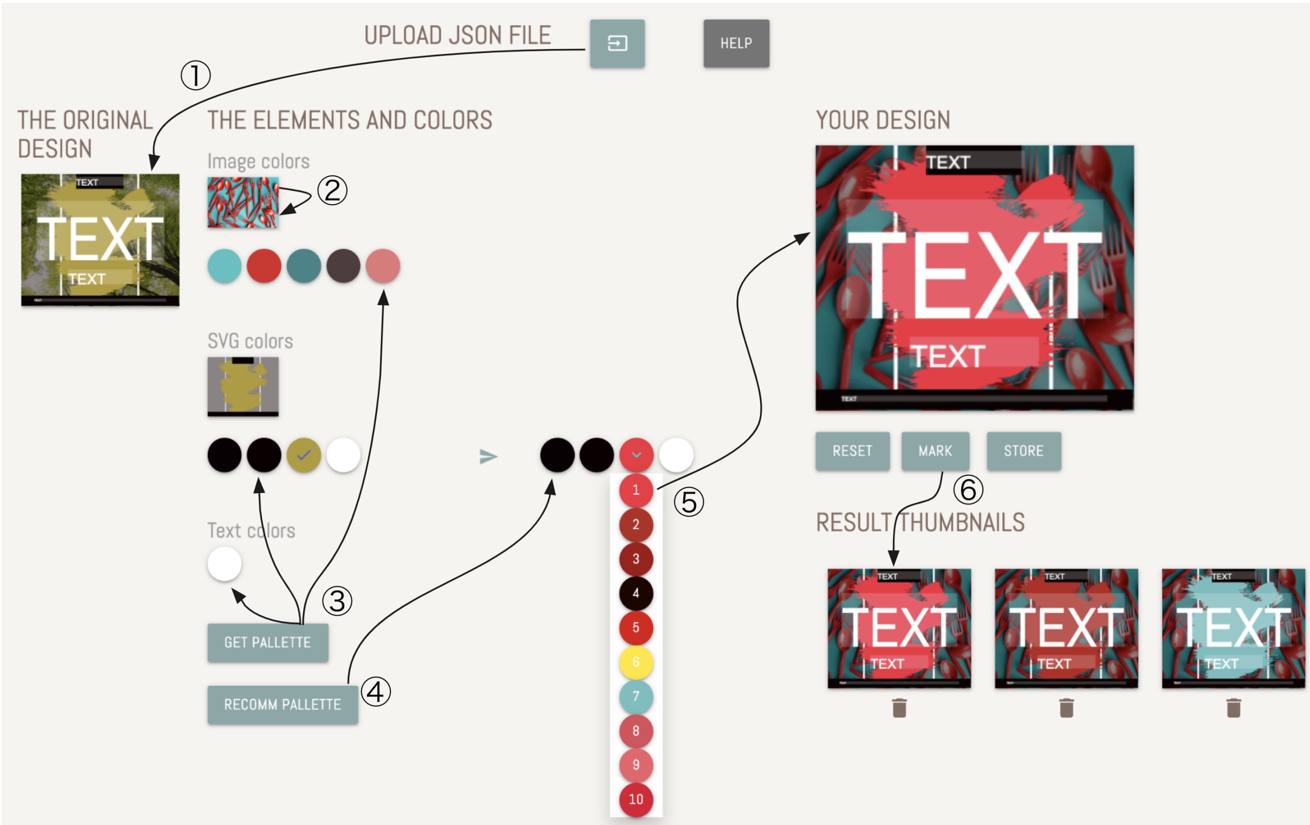

Figure 4: Interactive interface of color recommendation system for vector graphic documents contains six main operations: ① Input a JSON file. ② Replace image elements. ③ Get image-SVG-text palettes and select the colors for recoloring. ④ Get the recommended colors from the color recommendation engine. ⑤ Choose a recommended color and check the recolored result. ⑥ Mark the preferred results.

To compare our method with the Word2Vec-based model and the baseline model, we evaluate the accuracy and similarity of color prediction results by these three models. The comparison results of accuracy are shown in Figure 5, Table 2, and the comparison results of similarity are shown in Figure 6, Table 3. Our method with segment embeddings provides significantly better results than the Word2Vec-based method and the baseline model. The results show that the segmentation is effective in multi-palette representation learning and improves color recommendation performance. We suggest providing more than two color candidates in recommendation applications with high accuracy and users would like to find the desired color in the top N recommended colors.

Embeddings		Accuracy↑			
segment	position	@1	@3	@5	@10
w/	w/	0.27	0.45	0.53	0.63
w/	w/o	0.27	0.44	0.52	0.62
w/o	w/o	0.16	0.30	0.38	0.50

Table 1: Quantitative comparison of our model with and without segment and position embeddings on top N accuracy (N = 1, 3, 5, 10). Here is the mean value of 20 trained models.

Models	Accuracy↑				
	@1	@2	@3	@4	@5
Word2Vec	0.03	0.05	0.08	0.10	0.11
Ours w/o segment	0.23	0.32	0.39	0.43	0.46
Ours w/ segment	0.36	0.46	0.52	0.57	0.61

Table 2: Quantitative comparison of our models with and without segment embeddings and the Word2Vec-based model on top N accuracy (N = 1, 2, 3, 4, 5).

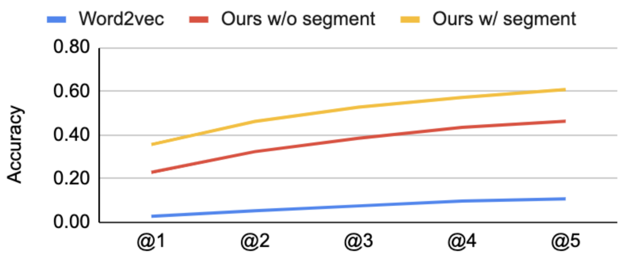

Figure 5: Quantitative comparison of our models with and without segment embeddings and the Word2Vec-based model on top N accuracy (N = 1, 2, 3, 4, 5).

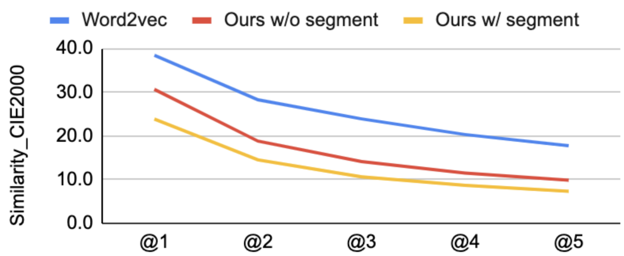

Figure 6: Quantitative comparison of our models with and without segment embeddings and the Word2Vec-based model on top N similarity (N = 1, 2, 3, 4, 5).

Models	Similarity↓ @1	@2	@3	@4	@5
Word2Vec	38.4	28.3	23.9	20.3	17.8
Ours w/o segment	30.6	18.8	14.1	11.5	9.9
Ours w/ segment	23.8	14.5	10.7	8.7	7.4

Table 3: Quantitative comparison of our models with and without segment embeddings and the Word2Vec-based model on top N similarity (N = 1, 2, 3, 4, 5).

4.2. Qualitative evaluation

Considering that color performance depends on human perception, we conducted a qualitative evaluation to verify the performance of the recommended results. We randomly selected 80 templates from the Crello test dataset. For each selected temple, one color in multi-palette which can be in image, SVG, or text element, is randomly masked for recoloring. The hue distribution of the masked color in 80 evaluation samples and all colors of multi-palette in the test dataset is shown in Figure 7. The neutral colors in some visually imperceptible elements are excluded during random selection, e.g. a text color with a tiny font size is ignored in this experiment. The recolored designs by the top 1 recommended colors of each model are shown in Figure 8.

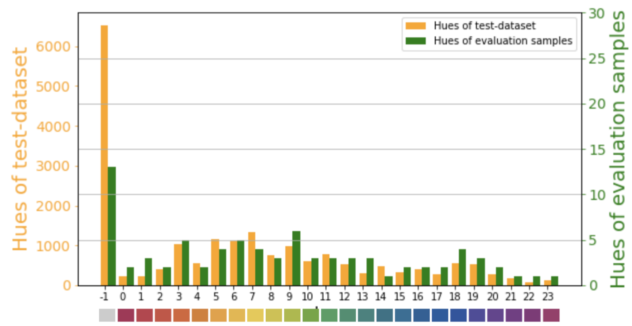

Figure 7: Hue distribution of selected colors in the evaluation samples and all colors in the Crello test dataset. Hue orders are based on the Practical Color Co-ordinate System, and we denote the neutral color as -1.

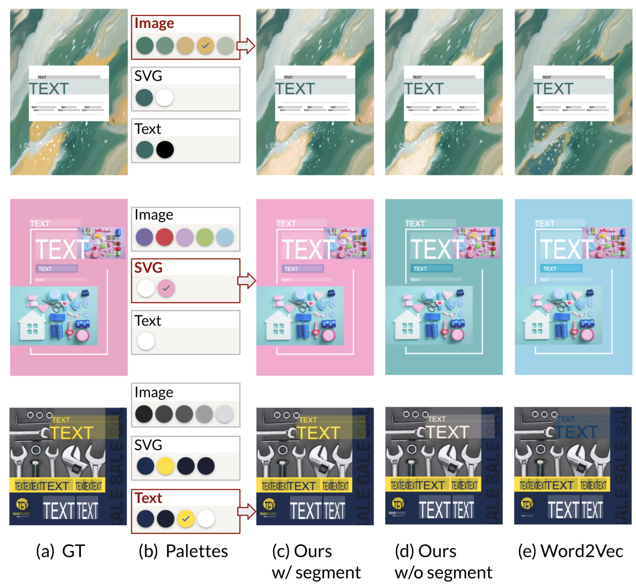

Figure 8: Color recommendation results with our proposed BERT-based models with and without segment embeddings, and the Word2Vec-based model. The three samples are recolored with one image color, one SVG color, and one text color.

We chose one original design (GT) and the top 2 recommended results from three models: our model with segment, the baseline model without segment, and the Word2Vec-based model. These seven designs are arranged together in an evaluation question. The participants are asked to select at most three good and three bad designs from the seven. We recruited 84 participants, with 68 non-designers and 16 graphic designers.

The evaluation results of good and bad design selections

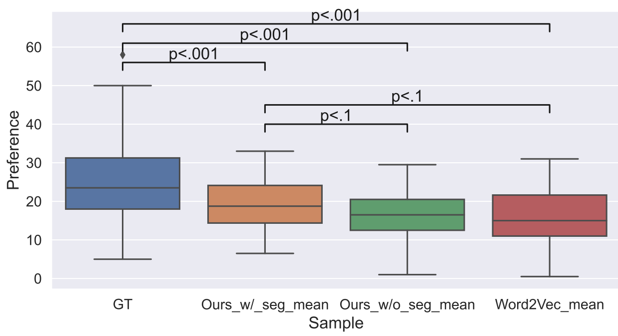

Figure 9: Evaluation results of the good designs from non-designers.

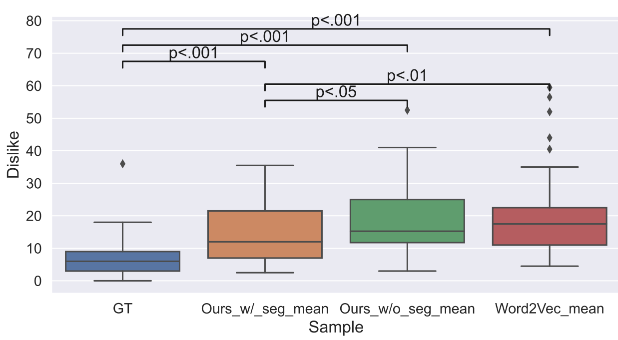

Figure 10: Evaluation results of the bad designs from non-designers.

from non-designers, are shown in Figures 9 and 10. The mean value of the top 2 recommendation results by the three models is shown. From the results, while our proposed model with segment embeddings performs worse than GT, it has a higher preference and lower dislike than the baseline model without segment and the Word2Vec-based model ($p < 0.1$). There is no significant difference between the baseline model and the Word2Vec-based model.

The evaluation results of good and bad design selections from designers are shown in Figures 11 and 12. The results are similar to the results from non-designers. Moreover, designers evaluate that our model with segment performs significantly better than the Word2Vec model (preference: $p < 0.001$, dislike: $p < 0.01$). From the evaluation results of non-designers and designers, the most prominent finding is that our proposed model performs better than the Word2Vec-based model. Moreover, segment embedding is necessary for the multi-paltte representation.

4.3. Interview study

To access the color recommendation system shown in Figure 4, we collected qualitative feedback from 12 professional designers aged 20-39. We provided a short tutorial of our system and asked the participants to explore color recommendations for recoloring one color in visual elements. There is more than one dominant color for the elements in some templates. We also asked participants to explore the recommendation of more than one color. Figure 13 shows a sample of recoloring three colors in SVG elements.

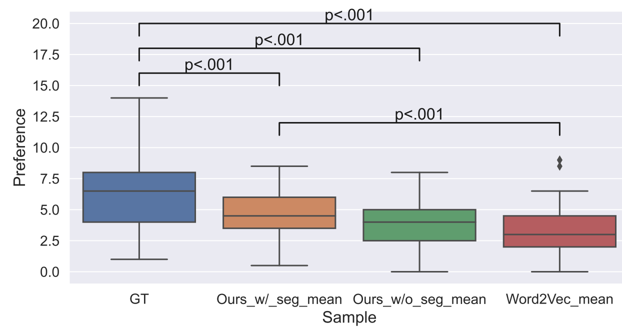

Figure 11: Evaluation results of the good designs from designers.

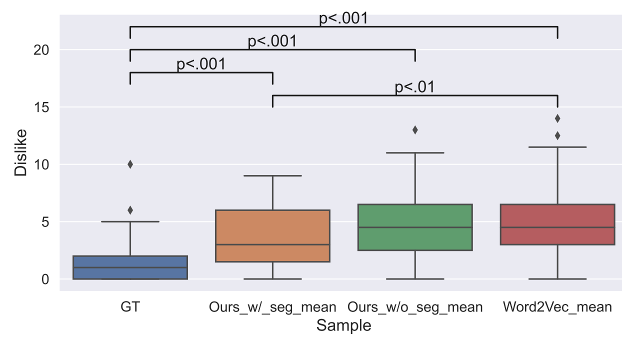

Figure 12: Evaluation results of the bad designs from designers.

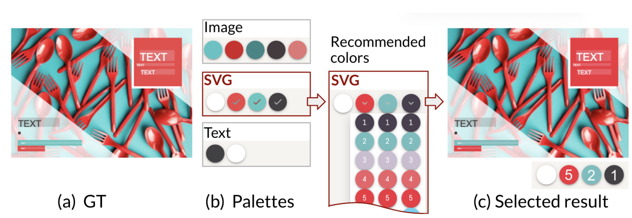

Figure 13: Color recommendation results for recoloring three colors in SVG elements. The number shown on the palette color is the recommended ranking.

Q1 Struggle with choosing colors in creative design works
Q2 Color recommendation tool is useful
Q3 This recommendation system is easy to use
Q4 Recommended results for one color look good
Q5 Recommended results for more than one color look good
Q6 Like to use this recommendation system for work

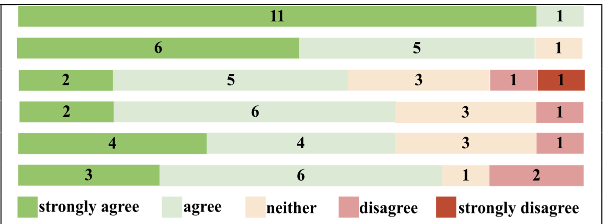

Table 4: Evaluation results from 12 designers in the interview study.

We prepared design templates and some image samples from the Crello test dataset. We asked participants to replace the image elements in design templates with image samples. The participants could also upload their images. The participants filled out a questionnaire after some trials. There were six questions, and each question had five choices: strongly agree, agree, neither, disagree, and strongly disagree.

The results are shown in Table 4. We observed the high demand for color recommendation systems in creative design works in Q1, and 91.7% participants answered that a color recommendation tool is useful for graphic design work in Q2. For Q3 and Q6, 58.3% participants answered that our current system is easy to use, and 75% participants would use our recommendation system for work. For the recommendation results, 66.7% participants were satisfied with the recommended colors by our system in one color and more than one color recoloring task as in Q4 and Q5. Generally, we received positive feedback for our color recommendation system and the recommended results from designers.

4.4. Limitations

Accuracy decreases when the masked color number increases. We generate the color sequence with a maximum of 15 colors and train our model by masking 10% of the token in each sequence. That means only one color is masked for prediction in most sequences in the training process. When the masked color number increases, the prediction accuracy decreases significantly in Table 5.

Masked colors	1	2	3	4	5
Accuracy@1↑	0.36	0.29	0.24	0.20	0.17

Table 5: Top 1 accuracy for predicting different numbers of masked colors.

Lack of diversity in recommended colors when more than one color is masked. Though users can freely combine the recommended colors and designers can get a satisfactory design by our recommendation system in the interview study, recommending a complete palette for each element by a learned model remains a problem. We generate the individual recommendations for each color and the recommended colors in the same element group are highly similar as in Figure 13. Furthermore, neutral colors have a semantic value to a color palette and we do not filter out these high-frequency colors. Neutral colors have a high frequency in the dataset as in Figure 7, and are more likely to be recommended than chromatic colors.

5. Conclusions

We proposed a masked color model for multi-palette representation to recommend colors for vector graphic documents and developed an interactive system of recoloring the specified colors in visual elements. The performance of the proposed approach was experimentally verified through quantitative and qualitative evaluations compared to the state-of-the-art method of a Word2Vec-based model and the baseline model. The color recommendation system received high evaluation from professional designers in interview study. Our method opens the door to recommend colors for vector graphic design based on a multi-palette of visual elements. We will explore to improve the performance of a complete palette recommendation and fine-turn our model for practical applications in future work.

References

[1] Adobe color. `https://color.adobe.com/`.

[2] Colourlovers. `https://www.colourlovers.com/`.

[3] Mahmoud Afifi, Brian L Price, Scott Cohen, and Michael S Brown. Image recoloring based on object color distributions. In *Eurographics (Short Papers)*, pages 33–36, 2019.

[4] Naofumi Akimoto, Huachun Zhu, Yanghua Jin, and Yoshimitsu Aoki. Fast soft color segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8277–8286, 2020.

[5] Hyojin Bahng, Seungjoo Yoo, Wonwoong Cho, David Keetae Park, Ziming Wu, Xiaojuan Ma, and Jaegul Choo. Coloring with words: Guiding image colorization through text-based palette generation. In *Proceedings of the european conference on computer vision (eccv)*, pages 431–447, 2018.

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword in-

formation. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[7] Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. Palette-based photo recoloring. *ACM Trans. Graph.*, 34(4):139–1, 2015.

[8] Ali Jahanian, Jerry Liu, Qian Lin, Daniel Tretter, Eamonn O'Brien-Strain, Seungyon Claire Lee, Nic Lyons, and Jan Allebach. Recommendation system for automatic design of magazine covers. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 95–106, 2013.

[9] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[10] 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*, pages 1488–1496, 2021.

[11] Eunseo Kim, Jeongmin Hong, Hyuna Lee, and Minsam Ko. Colorbo: Envisioned mandala coloringthrough human-ai collaboration. In *27th International Conference on Intelligent User Interfaces*, pages 15–26, 2022.

[12] Suzi Kim and Sunghee Choi. Dynamic closest color warping to sort and compare palettes. *ACM TRANSACTIONS ON GRAPHICS*, 40(4), 2021.

[13] Naoki Kita and Kazunori Miyata. Aesthetic rating and color suggestion for color palettes. In *Computer Graphics Forum*, volume 35, pages 127–136. Wiley Online Library, 2016.

[14] Kecheng Lu, Mi Feng, Xin Chen, Michael Sedlmair, Oliver Deussen, Dani Lischinski, Zhanglin Cheng, and Yunhai Wang. Palettailor: Discriminable colorization for categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):475–484, 2020.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[16] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Color compatibility from large datasets. In *ACM SIGGRAPH 2011 papers*, pages 1–12. 2011.

[17] Jianchao Tan, Jose Echevarria, and Yotam Gingold. Efficient palette-based decomposition and recoloring of images via rgbxy-space geometry. *ACM Transactions on Graphics (TOG)*, 37(6):1–10, 2018.

[18] Yili Wang, Yifan Liu, and Kun Xu. An improved geometric approach for palette-based image decomposition and recoloring. In *Computer Graphics Forum*, volume 38, pages 11–22. Wiley Online Library, 2019.

[19] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5461–5469. IEEE, 2021.

[20] Jie Yang, Yun Chen, Stephen Westland, and Kaida Xiao. Predicting visual similarity between colour palettes. *Color Research & Application*, 45(3):401–408, 2020.

[21] Xuyong Yang, Tao Mei, Ying-Qing Xu, Yong Rui, and Shipeng Li. Automatic generation of visual-textual presentation layout. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 12(2):1–22, 2016.

[22] Linping Yuan, Ziqi Zhou, Jian Zhao, Yiqiu Guo, Fan Du, and Huamin Qu. Infocolorizer: Interactive recommendation of color palettes for infographics. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

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