

Supplementary Material of Language-based Photo Color Adjustment for Graphic Designs

ZHENWEI WANG*, City University of Hong Kong, China
NANXUAN ZHAO*, Adobe Research, USA
GERHARD HANCKE, City University of Hong Kong, China
RYNISON W.H. LAU, City University of Hong Kong, China

Additional Key Words and Phrases: data-driven graphic design, photo recoloring, language-guided

ACM Reference Format:

Zhenwei Wang, Nanxuan Zhao, Gerhard Hancke, and Rynson W.H. Lau. 2023. Supplementary Material of Language-based Photo Color Adjustment for Graphic Designs. 1, 1 (April 2023), 9 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

A SYNTHETIC GRAPHIC DESIGN DATASET WITH INSTRUCTIONS

A.1 Design Elements Generation and Layout

As described in Sec. 4.1, we design our synthesis method based on both graphic design principles and design knowledge summarized from real-world graphic designs to better adapt our model to real design cases. Specially, we download 71 real-world graphic designs with diverse appearances from Canva¹. We summarize the distributions and statistics on: (1) the position and size of each element (e.g., the font sizes of title is larger than those of subtitle and plain text), (2) the number of elements in each class (e.g., a design often contains only a single title and one to three subtitles), (3) the number of lines, length of words, and font face in text-based elements (e.g., titles usually have no more than three lines and have a bolder fontface than other text), (4) the color contrast between text-based elements and shape or background for readability (i.e., avoid using the same color for overlapped elements), and (5) the appearance of shapes without content (e.g., rectangle, polygon).

We generate the design based on the order of: background color, shape, photo and text elements. The order of shape and photo can be swapped based on a ratio of 0.5. To harmonize the design, we follow the design principles in [Aland 2017; ANL [n. d.]; McGuire 2019; West 2020] by limiting the color palette of a design (except

the photo) in 3 to 5 colors, and assigning font faces based on both summarized distributions above and a font pairing method [Canva [n. d.]]. Please refer to our released code for more details.

A.2 Multi-granularity Instruction Generation

We generate the language-based instructions for each synthetic graphic design following a sentence template filling process. As the language-based instruction generally contains two parts, i.e., source color description and target region description, we first predefined two kinds of instruction patterns/templates to put these two parts of descriptions into a single sentence. While the first kind of instruction patterns puts the color description in front of the region description (e.g., “use [color description] to recolor [region description]”), the second kind of instruction patterns reverses the order of the two (e.g., “recolor [region description] into [color description]”). Here, different patterns use different conjunction words (e.g., using synonyms to replace “use” or “recolor”) to mimic diverse human expression habits. Please refer to our released code for more details on generating and merging the color/region descriptions.

B METHOD

B.1 Language-based Source Color Prediction Module

B.1.1 Granularity Recognition Subnet. In Eq. (1), both of \mathcal{G}_{cls} and \mathcal{G}_{att} are MLPs consisting of two fully connected layers and a softmax layer, whose weights are not shared with each other.

B.1.2 Class-aware Color Prediction Subnet. We introduce a class-aware voting strategy to obtain k base colors $C_b \in \mathbb{R}^{k \times 3}$ from the input design. We adopt max voting for text-based elements, and sum voting for filled-color-based elements according to the predicted class type by the granularity recognition subnet. Since the correct source colors for text-based elements are often at the center position, using max voting can help alleviate the disturbance from noisy background pixels. As filled-color-based elements generally cover a broad compact region, using sum voting to take dominant colors can avoid the disturbance of outliers (e.g., pixels in transition region).

B.1.3 Loss Functions. Since the predicted source colors have a random order, we use bi-partite matching [Kamath et al. 2021] to match each predicted source color with a corresponding confidence score to the closest ground truth source colors before calculating L_c and L_s in each training iteration.

*Both authors contributed equally to this research.

¹<https://www.canva.com/templates/>

Authors' addresses: Zhenwei Wang, zhenwwang2-c@my.cityu.edu.hk, City University of Hong Kong, Hong Kong SAR, China; Nanxuan Zhao, nanxuanzhao@gmail.com, Adobe Research, California, USA; Gerhard Hancke, gp.hancke@cityu.edu.hk, City University of Hong Kong, Hong Kong SAR, China; Rynson W.H. Lau, rynson.lau@cityu.edu.hk, City University of Hong Kong, Hong Kong SAR, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.

XXXX-XXXX/2023/4-ART \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

C EXPERIMENT

C.1 Implementation Details

C.1.1 Training details. To train the language-based source color prediction module in Sec. 5.1, we synthesize 20,500 design-instruction pairs using the method introduced in Sec. 4. We split the synthetic dataset by taking 20,000 for training and 500 for validating. To train the language-based image segmentation subnet for learning the initial region mask in the semantic-palette-based photo recoloring module (Sec. 5.2), we use the PhraseCut dataset with the formal training-validation split.

We use the Adam optimizer [Kingma and Ba 2014] with a learning rate $lr = 1e - 4$ for training. All the input designs and images are resized to 256×256 . When training the language-based source color prediction module, we use a curriculum training strategy by pretraining the element segmentation subnet and the granularity recognition subnet first to avoid model collapse, and then train the whole model jointly. We set the batch size as 10, the pretraining epoch as 20, and the epoch of jointly training as 20. The language-based image segmentation subnet is trained for 60 epochs before converging. We set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$ for balancing among loss terms. All the training and test processes are conducted on a PC with Intel i7-10700 CPU and a single RTX 3080 GPU. As for the non-training part in the semantic-palette-based photo recoloring module, we resize the input images and the initial region masks to 384 on the long side and keep the original aspect ratio for refinement and recoloring.

C.1.2 Network details. In the language-based source color prediction module, we leverage a state-of-the-art language-based image segmentation subnet [Feng et al. 2021] that uses an encoder fusion operation to obtain the fusion features f_{dl} , and generate the mask of element e , $M_e \in \mathbb{R}^{H \times W \times 1} \in [0, 1]$. Please refer to their paper and code² for more details about the segmentation subnet.

In the granularity recognition subnet, we first obtain the flattened 1D fusion feature f'_{dl} and then pass it into two different classifiers \mathcal{G}_{cls} and \mathcal{G}_{att} to obtain the predicted probabilities of element class p_{cls} and color attribute p_{att} , respectively. For both classifiers, we use an MLP that has a hidden layer with 512 neurons, batch normalization and rectified linear unit (ReLU) activation functions. The element classifier \mathcal{G}_{cls} has an output layer with 7 neurons (*i.e.*, 6 element classes and "None") and the color attribute classifier \mathcal{G}_{att} has an output layer with 12 neurons (*i.e.*, 11 colors and "None").

In the class-aware color prediction subnet, we first obtain the element class based on p_{cls} . Then, we use a class-aware voting strategy (Sec. 5.1.2) to obtain the top- k unique colors ($k = 10$) as the base colors $C_b \in \mathbb{R}^{k \times 3}$. We then represent each base color as a pure color map $B_i \in \mathbb{R}^{H \times W \times 3}$, $i \in [1, k]$, and concatenate all these color maps with the predicted element mask M_e and input design D to form a concatenated vector with 34 channels. Then we pass this concatenated vector into a class-aware source color refinement subnet (*i.e.*, selected from two subnets with same architecture but with different parameters) to obtain the color residuals C_r and confidence scores s_i . In the source color refinement subnets, we use 4

convolutional layers with batch normalization and ReLU activation functions, followed by 4 global average pooling layers, to convert the concatenated vector into 4 flattened 1D features under different scales. Finally, we concatenate these 1D features and pass them into an MLP with 2 hidden layers with 1024 and 512 neurons, respectively, and an output layer with $4 \times k$ neurons (*i.e.*, 3 channels for RGB values and 1 channel for color confidence score). Please refer to our released code for more details of the network architecture.

In the semantic-palette-based photo recoloring module, we use the same image segmentation subnet as introduced above to obtain the fusion features f_{il} , and generate the initial region mask $M_r^{init} \in \mathbb{R}^{H \times W \times 1} \in [0, 1]$. The rest of the semantic-palette-based photo recoloring module is optimization-based and thus does not need to be trained.

C.2 More Recoloring Results of Our Method

Fig. 20 shows more results on language-based recoloring for various graphic designs with photos under diverse appearances.

²<https://github.com/fengguang94/CEfNet>



Fig. 20. Our results on language-based photo recoloring on graphic designs. For each design case, we show the original design on the left and our result(s) on the right, with the predicted source color and the input instruction below. We highlight the words related to the source colors and target regions in bold. For a single design case, our model allows users to specify different source colors and target regions through instructions, as shown in the rightmost column.



Fig. 21. Comparison results with ReColGAN [Khodadadeh et al. 2021]. Our results are in the right most column.

C.3 Comparison with Existing Works

C.3.1 Comparison Results with ReColGAN and Imagic. As discussed in Sec. 6.1, we also compare our method with two state-of-the-art text-guided image editing method: ReColGAN [Khodadadeh et al. 2021] and Imagic [Kawar et al. 2022] that support structure-preserved recoloring. To fit our problem, we manually convert our instructions to the prompts in a format used by their work. For example, we use "recolor flower to orange" and "recolor dress to blue" for ReColGAN and "a photo of a blue car" and "a photo of a red chair" for Imagic.

Note that, without the released code and pretrained models, we can only compare with results in the original papers, as shown in Fig. 21 and Fig. 22. As can be seen, by relying on pretrained object detectors and saliency mask segmentation networks with predefined color tags, ReColGAN cannot recolor with accurate color, and may generate color-bleeding artifacts with imperfect segmentation masks. With the advantage of powerful pretrained text-to-image diffusion models (*i.e.*, Imagen[Saharia et al. 2022]), Imagic can produce high-quality recoloring results that semantically align with the input text. However, as the target image is obtained by applying a linear interpolation in the text embedding space, original details and object structure may be changed (*e.g.*, the height of the chair in the 2nd row). In addition, the source colors may not be accurate as in the design itself with the vague color words.

C.3.2 User Study Setup. As discussed in Sec. 6.1.4, participants were invited to complete a questionnaire consisting of 30 pairwise comparisons in person, using the same color-calibrated screen. For each comparison, we show a single design-instruction pair and present our result with the others (*i.e.*, either from Open-edit, ManiGAN, or the ground truth/designers) side by side in random order. For each comparison, we ask the participants two questions: "Which recolored photo looks more natural?" and "Which recolored photo is

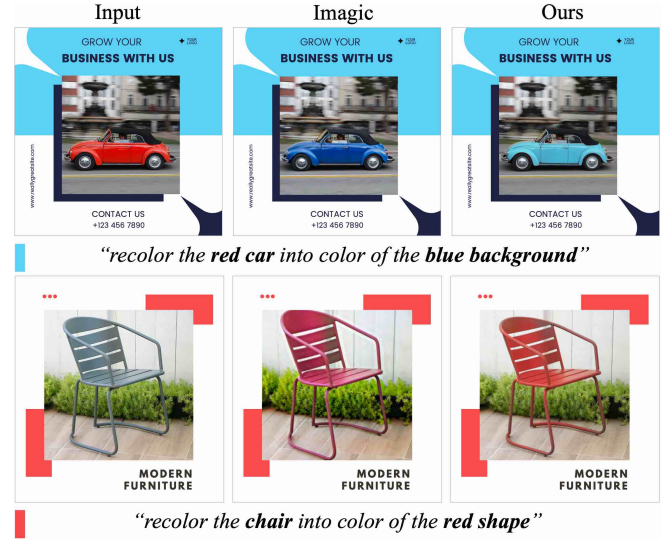


Fig. 22. Comparison results with Imagic [Kawar et al. 2022]. Our results are in the right most column.

more faithful to the given instructions?" in a two-alternative forced-choice (2AFC) manner to better measure small differences [Zhao et al. 2018]. Each participant can only work on at most 2 questionnaires and each pair of comparisons is evaluated by at least 3 different participants.

C.4 The Effect of Language-based Source Color Prediction Module

C.4.1 Quantitative Results on Confidence Scores Evaluation. As shown in Fig. 10 (b), a smaller threshold can include more noisy predictions, reducing the accuracy. But if the threshold is too large, there may only exist a single result, failing to match the sample with multiple source colors. To avoid an empty prediction, we keep the source color with the largest confidence, thus the line becomes flat after a threshold of 0.75.

C.5 The Effect of Semantic-palette-based Photo Recoloring Module

C.5.1 Baselines. We design the baselines as follows: 1) **Scribble-based recoloring** [Levin et al. 2004]. We use the source colors to draw scribbles within the target regions and original colors for the remaining regions. 2) **Deep user-guided recoloring** [Zhang et al. 2017]. The color points are added in two different modes (*i.e.*, Sparse and Dense). In the Sparse mode, the hint points are randomly sampled within the target regions, and in the Dense mode, the hint points are sampled per-pixel within the target regions. 3) **Semantic-based photo recoloring** [Zhao et al. 2021]. It proposes a recoloring network for propagating colors within the given binary local region. We use the initial region mask and source colors predicted by our model as the inputs to the recoloring network. This baseline can be regarded as an ablation variant of our model without the semantic-palette-based region refinement. 4) **Palette-based photo recoloring** [Tan et al. 2018]. We modify the colors within the palette

that are dominant in the initial region mask predicted by our model to the source color.

C.6 The Effect of Using the Language-based Input

We conduct a usability user study [O'Donovan et al. 2015; Xiao et al. 2021] to further evaluate the effectiveness of our language-based input modality in helping novices to recolor photos in the context of graphic designs. The usability user study results demonstrate that our language-based input modality significantly speeds up the authoring process compared with a traditional scribble-based interface. In practice scenarios, users can quickly brainstorm ideas and generate diverse satisfying results within 1 min on average using our method.

C.6.1 Setup. We invited ten novice users to test two interfaces: 1) Language-based interface of our *LangRecol* framework (**Ours**); and 2) a traditional scribble-based interface without language as input (**Baseline**). Although some techniques like instance segmentation [Cheng et al. 2022] or semantic segmentation [Li et al. 2022] may ease the mask creation process, they mainly focus on semantic objects (e.g., a person) rather than semantic color regions (e.g., clothes). Users still need to learn and interact with the system using these techniques. We thus adopt sketching for the baseline interface, so that the user study is more intuitive. As the usability study focuses on evaluating different input modalities, for fair comparisons, both interfaces adopt our semantic-palette-based photo recoloring module for producing results. More details of the step-by-step usage of these two interfaces can be found in Sec. C.6.2.

Before starting the study, users took a 15-min tutorial, including getting familiar with the instructions and basic element types, drawing initial region masks, and selecting source colors by color pickers, etc. For each user, we first randomly selected 5 design-instruction pairs from the test set as the recoloring tasks and asked the user to recolor following the instructions. To reduce bias, the order that users test the two interfaces was set to random. Besides, when testing the second interface, we reassign the previous 5 tasks selected by each user to another user so that each user performs different tasks when testing different interfaces, and each task can be tested on both interfaces. There were no time limits, and users could stop once they felt that the results were acceptable or could not be improved further.

During the study, the time usage for each design case is recorded. After conducting the design tasks, participants were asked to provide a five-point System Usability Scale (SUS, 1 = strongly disagree to 5 = strongly agree) on four different perspectives for each interface, including (1) the usability of the interface (i.e., a higher score indicating that the interface is easier to use in practice), (2) the faithfulness of the results to the task instruction (i.e., a higher score indicating that the results are more consistent with the task intention), (3) the efficiency of finishing an editing task (i.e., a higher score indicating that users can obtain satisfactory results in less time), and (4) the overall preference of the interface (i.e., a higher score indicating that the users prefer the current interface over another).

C.6.2 Step-by-Step Usage of the Interface. Fig. 23 and Fig. 24 demonstrate the step-by-step usage of our language-based interface and a traditional scribble-based interface. When testing the language-based interface, if users are not satisfied with specific results, we allow them to refine the results freely by either manually editing the target region mask and source colors based on the initial predictions, as shown in Fig. 23 (d)~(f), or adjusting the input instructions. More live demonstrations can be found in the supplementary video.

C.6.3 More User Feedback. Many users preferred our novel language-based interface to the baseline and provided many positive feedback comments on our interface, such as "This interface is really cool as I can create diverse (recoloring) designs with only a single click.", "I enjoy this creative interface, where I can directly tell the system what I want via typing or even voice in the future.", and "This interface is efficient and practical, especially in the initial stages of creative expression and brainstorming." Some users appreciated the region refinement function: "This tool has the best image matting function I have ever seen. I can specify nearly arbitrary objects, and the refined region mask also looks good." Besides, a user preferred the baseline interface as it is more controllable: "I enjoy the feeling of drawing regions manually since I can control more details and obtain more reliable results." Further, users also provided constructive suggestions, such as "I believe the best interface is a combination of these two." and "What appealed to me most is that while the language interface cannot be completely accurate (the accuracy was about 90% for me), it is accurate enough for the initial content selection. Thus, combining with only a small amount of manual fine-tuning can bring me the desired result, which is fantastic and can greatly reduce the workload."

D MORE EXAMPLES RESULTS ON APPLICATIONS

Fig. 25 to Fig. 28 show more example results on applications introduced in Sec. 7.

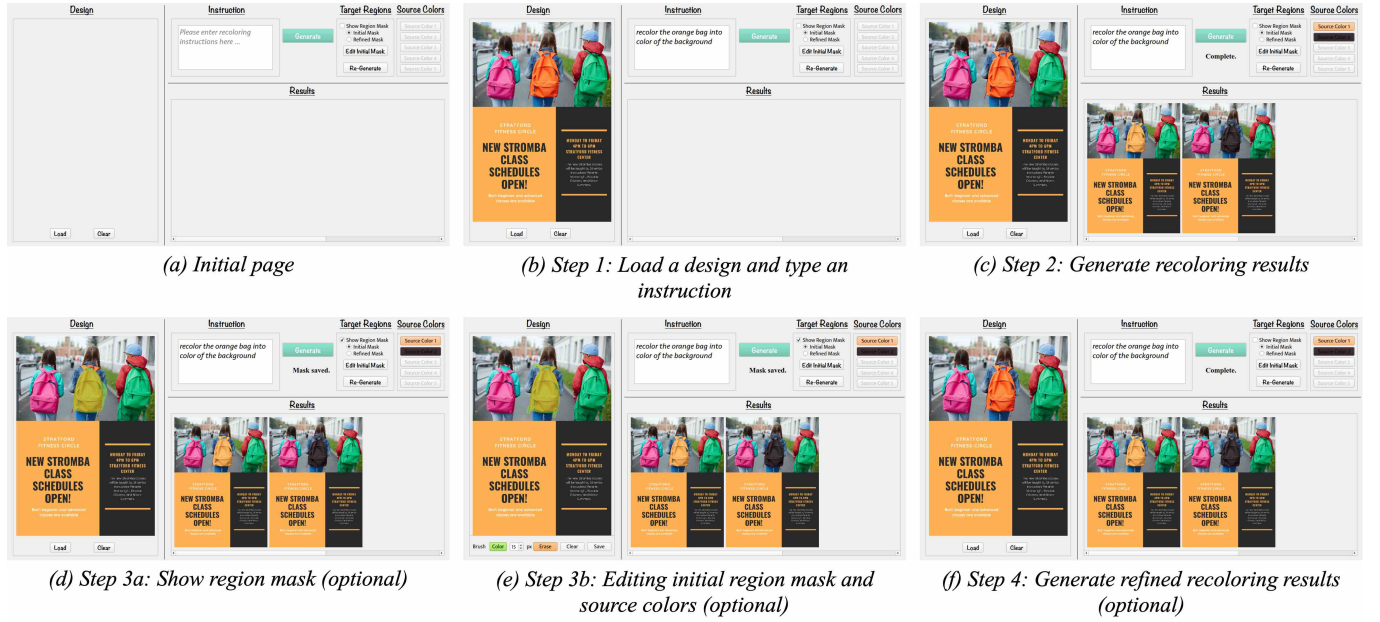
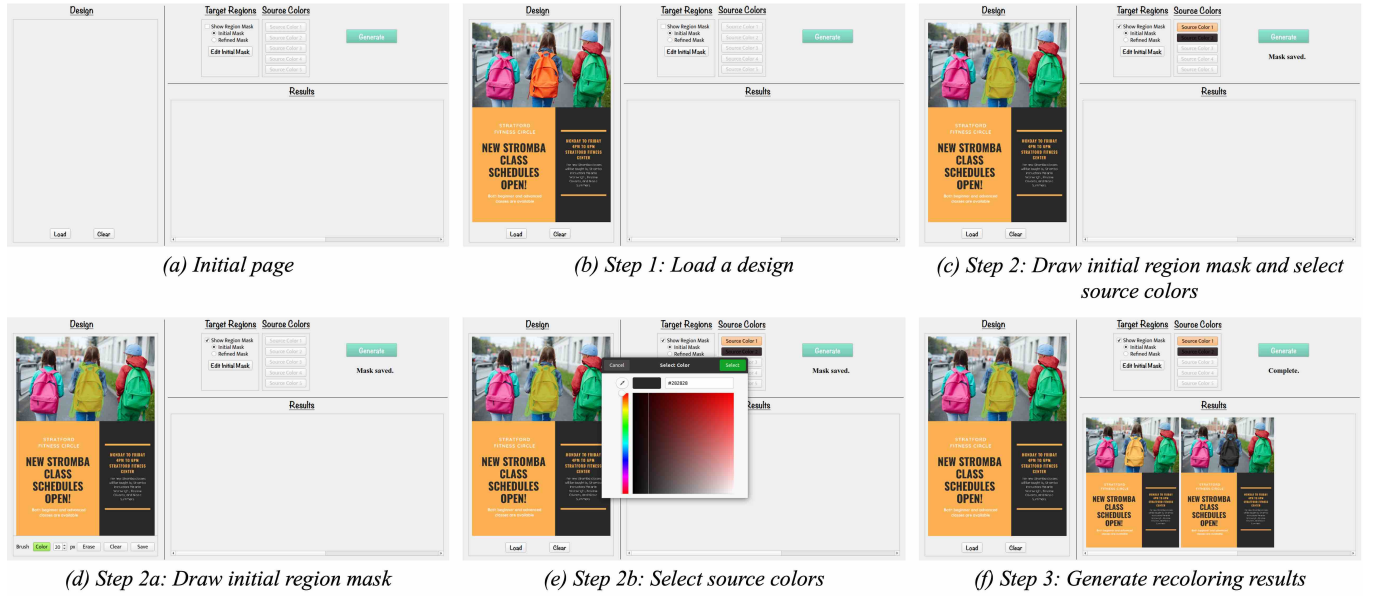
Fig. 23. Step-by-step usage of the language-based interface of our *LangRecol* framework (**Ours**).Fig. 24. Step-by-step usage of the traditional scribble-based interface without language as input (**Baseline**).



Fig. 25. Design template pairing. Our model can adjust photo color based on a set of design templates for users to quickly brainstorm or finalize the template.



Fig. 26. Brochure Photo Recoloring. Our model can recolor multiple photos with a single design.

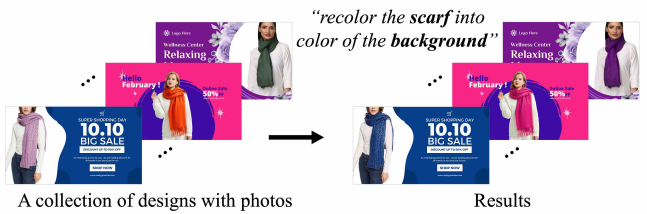


Fig. 27. Recoloring a design collection. We show a design case for banners used in online promotion.



Fig. 28. Iterative graphic design photo recoloring.

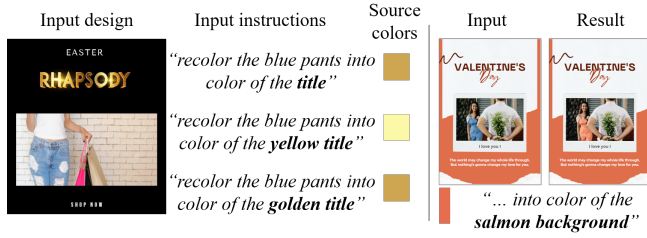


Fig. 29. Left: our language-based source color prediction results on a graphic design with textured title. Right: our language-based photo recoloring result using an unseen color name “salmon”.

E MORE DISCUSSIONS ON THE SCOPE OF OUR WORK

Apart from the limitations discussed in Sec. 8, we further discuss the scope of our work here.

First, our model can deal with relatively complex cases, such as irregular layouts and various aspect ratios (Fig. 16 and 26), and photograph image background (“HAPPY Easter” poster and “Welcome Spring” poster in Fig. 25). For elements with gradient colors, *e.g.*, textured titles, our model may only predict a single representative color, and the color can be further customized by adding color attributes, *e.g.*, “yellow” or “golden”, in the instruction, as shown in Fig. 29 left.

Second, even though we have tried to synthesize instructions as human-like as possible by considering diverse instruction patterns, it is still hard to capture all variations of real-world human instructions. In general, the users need to follow our instruction patterns by providing two parts of information (*i.e.*, descriptions of source color and target region). Simple grammar errors are tolerable, but instructions that significantly deviate from our patterns will likely fail, such as “help me recolor the bag so that it looks more fashionable”. We may consider alleviating this problem in the future by fine-tuning our model with real-world user instructions.

Third, for more intuitive usage by novices, we only consider 11 basic color attributes (*i.e.*, blue, brown, green, orange, pink, purple, red, yellow, black, grey, and white) when synthesizing the instructions for training, as stated in Sec. 4.2. Although our model is trained with limited color words, it has a large vocabulary and each word is represented as a word vector [Mikolov et al. 2018] (*i.e.*, similar color words have shorter distances), which naturally enables our model to understand color words outside of the training data. For example, considering the third case in the first row of Fig. 7, we can still obtain a correct result after replacing “orange” with an unseen color word “salmon”, as shown in Fig. 29 right.

Fourth, due to the inherent ambiguity of natural language, it might not be easy for users to give a precise description in some specific scenarios. When there are multiple similar objects, our method allows users to give a fine-grained description to specify the target object, *e.g.*, using color attributes in Fig. 13 (*i.e.*, “orange bag”) to distinguish different bags. However, for even more complex scenes, *e.g.*, multi-objects with exactly the same appearance, language-based instructions alone are not enough to tell the target object from others. In these scenes, incorporating a small amount of manual fine-tuning based on prediction can be a solution (Fig. 23(d)-(f)), as discussed in Sec. C.6.3.

F POTENTIAL ETHICS ISSUES

Like two sides of a coin, while we have proposed an intuitive and easy way to help designers and novices recolor photos of single-page graphic design, it may carry a negative side and bring potential ethical concerns, especially in cases where copyrighted designs are manipulated without permission. We suggest that our current system can be used only for non-commercial purposes. Users should not edit any copyrighted designs without explicit permission from the owner. We also encourage the research community to further discuss mechanisms for preventing abuse in the future.

REFERENCES

- Maggie Aland. 2017. *10 tips and ideas to make a flyer that stands out*. Retrieved Jan 03, 2022 from <https://www.lucidpress.com/blog/10-creative-ways-to-make-flyer-stand-out>
- ANL [n. d.]. *Guide to Effective Poster Design*. Retrieved Jan 03, 2022 from <https://www.anl.gov/education/guide-to-effective-poster-design>
- Canva [n. d.]. *The best Google Font combinations to try*. Retrieved May 06, 2022 from <https://www.canva.com/learn/best-google-font-combinations/>
- Tianheng Cheng, Xinggang Wang, Shaoyu Chen, Wenqiang Zhang, Qian Zhang, Chang Huang, Zhaoxiang Zhang, and Wenyu Liu. 2022. Sparse instance activation for real-time instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4433–4442.
- Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. 2021. Encoder fusion network with co-attention embedding for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15506–15515.
- Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. MDETR-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1780–1790.
- Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2022. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276* (2022).
- Siavash Khodadadeh, Saeid Motiian, Zhe Lin, Ladislau Boloni, and Shabnam Ghadar. 2021. Automatic Object Recoloring Using Adversarial Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1488–1496.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- Anat Levin, Dani Lischinski, and Yair Weiss. 2004. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*. 689–694.
- Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. 2022. Language-driven Semantic Segmentation. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=RrDjddCLN>
- Sara McGuire. 2019. *Poster Design Guide: How to Make an Eye-Catching Poster in 2020*. Retrieved Jan 03, 2022 from <https://venngage.com/blog/poster-design/>
- Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhresch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.
- Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. Designscape: Design with interactive layout suggestions. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 1221–1224.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487* (2022).
- Jianchao Tan, Jose Echevarria, and Yotam Gingold. 2018. Efficient palette-based decomposition and recoloring of images via RGBXY-space geometry. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.
- Chloe West. 2020. *The Ultimate Guide to Flyer Design*. Retrieved May 02, 2022 from <https://visme.co/blog/flyer-design/>
- Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. 2021. Sketch-HairSalon: Deep Sketch-based Hair Image Synthesis. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2021)* 40, 6 (2021), 1–16.
- Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. 2017. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999* (2017).
- Nanxuan Zhao, Ying Cao, and Rynson WH Lau. 2018. Modeling fonts in context: Font prediction on web designs. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 385–395.
- Nanxuan Zhao, Quanlong Zheng, Jing Liao, Ying Cao, Hanspeter Pfister, and Rynson WH Lau. 2021. Selective Region-based Photo Color Adjustment for Graphic Designs. *ACM Transactions on Graphics (TOG)* 40, 2 (2021), 1–16.