CoColor: Interactive Exploration of Color Designs

Lena Hegemann, Niraj Ramesh Dayama, Abhishek Iyer, Erfan Farhadi, Ekaterina Marchenko, Antti Oulasvirta
firstname.lastname@aalto.fi
Department of Information and Communications Engineering, Aalto University
Espoo, Finland

ABSTRACT

Choosing colors is a pivotal but challenging component of graphic design. The paper presents an intelligent interaction technique supporting designers' creativity in color design. It fills a gap in the literature by proposing an integrated technique for color exploration, assignment, and refinement: CoColor. Our design goals were 1) let designers focus on color choice by freeing them from pixel-level editing and 2) support rapid flow between low- and high-level decisions. Our interaction technique utilizes three steps – choice of focus, choice of suitable colors, and the colors' application to designs – wherein the choices are interlinked and computer-assisted, thus supporting divergent and convergent thinking. It considers color harmony, visual saliency, and elementary accessibility requirements. The technique was incorporated into the popular design tool Figma and evaluated in a study with 16 designers. Participants explored the coloring options more easily with CoColor and considered it helpful.

KEYWORDS

color design; design exploration; interaction technique; creativity support tools

ACM Reference Format:
Lena Hegemann, Niraj Ramesh Dayama, Abhishek Iyer, Erfan Farhadi, Ekaterina Marchenko, Antti Oulasvirta. 2023. CoColor: Interactive Exploration of Color Designs. In *28th International Conference on Intelligent User Interfaces (IUI '23), March 27–31, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3581641.3584089

1 INTRODUCTION

Colors play a key role in the perception of a design. They guide visual attention [25], evoke associations [15] and emotions [32], are critical for accessibility [3], and speak to aesthetics [17]. For designers, deciding on good colors is far from trivial. The designer must choose and place colors such that all of these effects support the purpose of the design at hand. For instance, a poster advertising a tasty food should look aesthetic, render all relevant information readily visible to the target audience, and stress tastiness. Exploration is important in the process of achieving this. Because of the multitude of requirements, which might even show mutual conflict, and the numerous ways in which colors can be combined and assigned to a given sketch, it is vital that designers be able to consider a large number of options, iteratively [6].

With this paper, we elaborate on an intelligent interaction technique and computational approach developed to support rapid exploration of color designs. The work applied a structured approach informed by scholarly understanding of color design. Proceeding from prior work on color design and art [20, 22], we assumed that designers operate via three steps: they extract color themes from source materials, focusing on objects and salient areas [27]; create palettes; and apply colors to the design at hand. The process is iterative – seeing how a color palette works when applied to the design can change the course of one's thinking and spark new ideas of possible avenues. Hence, designers' rapid movement back and forth between these steps should be supported. Consequently, our approach supports exploration in three conceptual design spaces and the transitions between them during color-design: a 1) focus space, 2) palette space, and 3) colorization space. The aim behind the design activity in progress drives the choice of focus. In the example of advertising a tasty food, the notion "tasty" or an image that the design ought to highlight might serve as the focal object. In the shift to the next space, this focus informs the choice of colors. For this step, designers often seek visual cues for translating the description of the focus into visual form [49] or extract colors from the focal image [22].

For highlighting the focus, the emerging design benefits from combining reference colors in the most relevant parts of the image [22] with suitable colors that match with these to create visual interest and contrast [32]. Finally, in the colorization space, colors are applied (assigned) to the layout. Visual inspection of the colorization may inspire adaptation of the layout or of the color choices, thereby prompting further iterations in the color-design process. Even the choice of focus might well change. For instance, attention to green tones could prompt a shift of the focus to the healthful aspect of the tasty food.

Most tools for color design enforce separating the three steps in the exploration process. Some systems make suggestions for palettes [30, 39], advanced color pickers apply colors to artists' digital paintings or images [31, 42–44], etc. Each covers only part of the process. Meanwhile, no fully automated systems [8, 9] support interaction at intermediate stages. Clearly, current tools do not support rapidly iterating.

We address this gap by proposing CoColor[1], a novel intelligent interaction technique for color-schema design. Backed by a three-pronged computational approach, it facilitates iteration by supporting 1) moving to the next step, through generating palettes from

[1]Project page at `https://userinterfaces.aalto.fi/cocolor`

IUI '23, March 27–31, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0106-1/23/03.
https://doi.org/10.1145/3581641.3584089

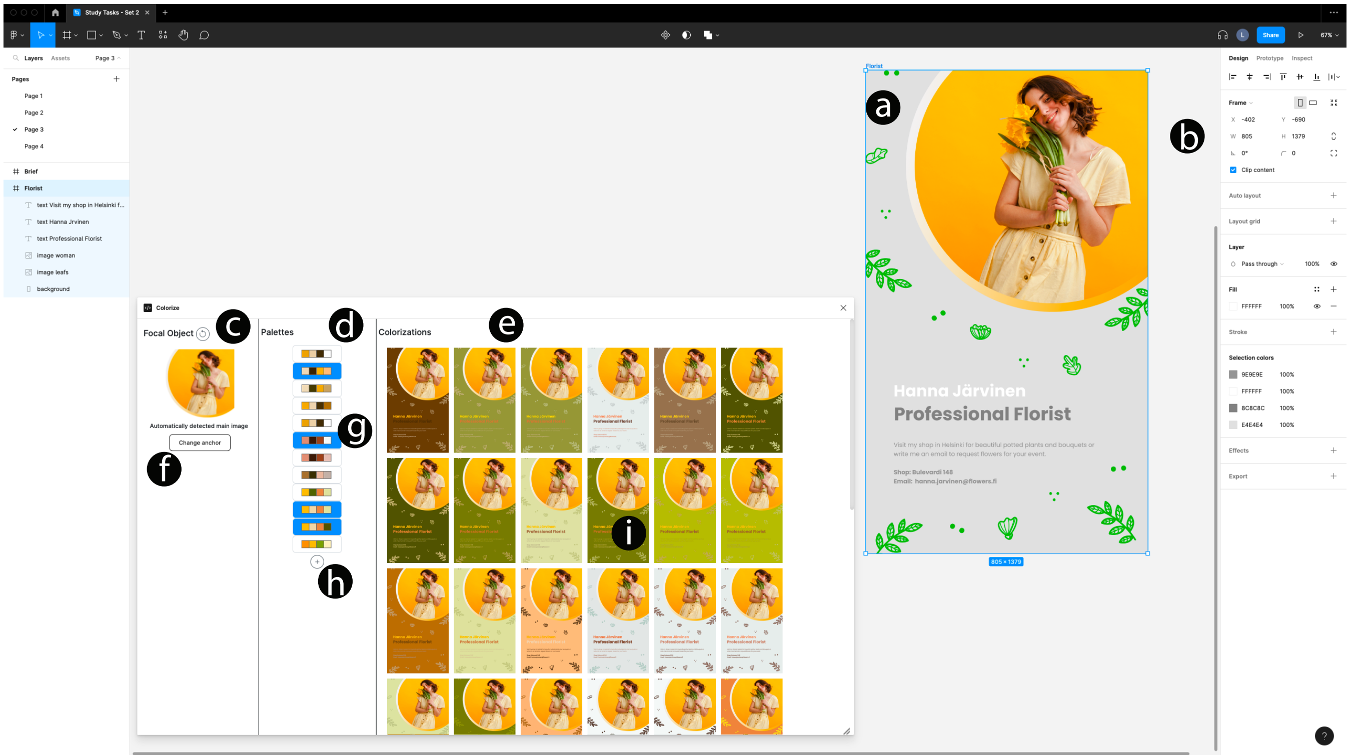

Figure 1: CoColor is an interaction technique that assists with the coloring of visual designs. It builds on a conceptual approach to color design that follows three distinct steps in color exploration: 1) choice of focal object, 2) choice of color palette, and 3) assignment of the palette to the design. Here, the technique is implemented for the popular design tool Figma: The designer opens the Figma plugin and selects the frame for co-coloring (a). CoColor's interface presents a section showing the image given focus (c), a list of palettes generated by various methods (d), and a preview of several new colorized designs (e). Users can switch from the automatically detected focal image and influence which elements to color (f), and they can filter the results by palette (g) or add palettes (h). Clicking a color suggestion (i) renders a new frame with the chosen color on the canvas (b). Image Source: example design from Freepik.

focal objects (via state-of-the-art saliency models) and colorizing the design case (by using palettes); 2) allowing designers to input changes and additions within every step; and 3) suggesting multiple high-quality outputs in every step, for selection and inspiration. For this paper, the interaction technique was implemented as a plugin for the popular Figma design tool; Figure 1 depicts our interaction technique's operation in that setting.

CoColor employs the following flow: It begins by enabling the designer to designate a focal object for the design (in the plugin context, a graphical element or image). After CoColor received the design and focal object, it detects the other elements of the layout (size, location, text, etc.), and initiates the coloring process by extracting colors from the focal object. Then, supplementary matching colors are generated to complete the palette and afford interesting contrasts. From each palette generated, the system offers several potential color assignments, to ensure the designer's control over every coloring decision and provide fine-grained decision over which aspects/options to explore next. The designer may opt to explore the generated palettes and corresponding colored designs next, iterate over the palettes to obtain more options, or select and edit one of the designs offered. A designer can arrive at the final color scheme more easily when provided with color palettes and shown how they look when applied to the design in question.

The paper represents several contributions:

- An intelligent interaction technique for color design. Rather than automate coloring, it helps designers explore color spaces rapidly. The approach builds on a three-step process for iteratively picking relevant colors on the basis of focal objects and applying colors to the graphical design.
- Integration of CoColor into a real-world design tool as a Figma plugin, backed by a full intelligent-coloring pipeline. This extends from picking the focal color, through generation of relevant palettes and assignment of colors to visible elements within a given layout, to final rendering of compelling and relevant colored designs, with auxiliary capabilities to ensure legibility.
- An in-depth controlled study attesting that participating designers found CoColor useful, interacted in every design space supported, and explored their coloring options with greater ease.

2 RELATED WORK

Established means of colors' extraction from images, colorization methods, and tools for color picking provided the backdrop for our work. These are summarized below.

2.1 Extraction of Colors from Images

2.1.1 Color Quantization. Color quantization is the process that represents an image via a smaller set of colors. Among the common methods for this mapping are clustering algorithms such as *k*-means [50] and a modified median cut [1] over the pixels of an image in the chosen color space. One use case for color quantization is colors' extraction from inspirational materials, whether art [39], any kind of image [21, 24], or mixed media [18]. Developing a model grounded in a study of how humans perform color quantization, Lin and Hanrahan [27] verified that people tend to focus on salient areas and populate their palettes of extracted colors with diverse colors. Also, the palettes extracted from the salient areas received higher ratings for their aesthetics. These findings encouraged us to apply state-of-the-art saliency models to focal images before quantization. While color quantization serves picking a set of colors from an image, the set of colors already present in the given image may not suffice for a palette rich enough to meet a color design's numerous requirements, from highlighting a focal image to dealing with the various other elements of a layout.

2.1.2 Detection of Salient Areas. An automated analogue of the process by which humans manually choose colors from salient parts of images [27] requires computational saliency models. Many state-of-the-art models employ deep-learning methods: DeepGaze I [26] predicts fixation locations from the layer-specific outputs of an already trained object-recognition model. Its approach uses a linear combination of these outputs, blur, and a prior distribution to incorporate center bias (our tendency to look at the center of an image first). With DeepGaze IIE's advances [29], the developers combined several object-detection models, to overcome the problem of any single model's limited generalizability to different datasets. The state-of-the-art model U2-net [40] is trained from scratch from a suitable dataset for a slightly different task: detecting salient objects. The goal is not to predict fixations but to isolate the most salient object entirely from the rest of the image. While the much more lightweight U2-netp outputs similar results, it misses parts of the object more often [40]. Other projects pursuing speed benefits have turned to theory-based models rather than deep learning. For instance, the "spectral" algorithm [33], following the assumption that redundant areas are of lesser interest, favors minimal redundancy of information in the areas suggested. The "fine grained" algorithm [12], in turn, applies a retina-based model to calculate saliency in terms of dark areas surrounded by light ones and *vice versa.* A key advantage over the spectral approach lies in providing very detailed saliency maps. Figure 2 presents two example outputs for each of the aforementioned models we explored, both deep-learning- and theory-based.

2.2 Identification of Matching Colors

Scholars have proposed several approaches for generating color palettes. Patterns in color spaces such as relations of chroma, lightness, and hue frequently fuel generation of color themes. For instance, palette design in research [13, 30, 48] and publicly released tools[2] alike regularly apply templates to ensure that the colors in the set are at specific angles to each other in a color wheel (a circle representing a spectrum with similar hues adjacent [17, 34]). In contrast, other methods learn palette models from datasets. These have found preferences for cyans, warm colors, and gradients but not for any color-wheel angles [36]. Another data-based model, by Kim and Suk [21], adjusts the lightness and saturation of a given hue to match the desired mood. This was inspired by an empirical model that predicts the popularity of two-color combinations via chroma, lightness, and hue calculations [37]. Though models for color palettes' aesthetics, whether data- or theory-based, aid in finding colors that match each other and data-based models hold promise for generation of realistic palettes, those available today are trained and evaluated from palette colors alone, without context.

2.3 Color Assignment

The next step is to assign colors from the palette to the elements of the design. Applying a probabilistic approach [28] and the aforementioned harmony model [37] in combination with Gestalt-laws-based rules [23] has yielded aesthetically pleasing graphical patterns, but these are limited to decorative shapes. Looking beyond patterns, Gu and Lou [8] developed a generative design engine to apply color palettes to web layouts by means of data-based models of contrast, harmony, and semantics. The task of colorization is closely related to color assignment, and there are many approaches to the numerous use cases for assigning colors to grayscale images in computer graphics [14]. The ones tying in most strongly with our work are semi-automatic methods such as transferring color from other images or from written descriptions, often via color palettes [14]. Colorization-based assignment for drawings, comics/manga, and photographs [14] all have produced realistic and pleasing results; however, each of these is a specialist domain, limited to specific use cases. In contrast, we sought versatility, to support designers with many forms of graphic, websites, and user-interface (UI) design.

2.4 Color-Design Tools

Many authors have proposed support tools for coloring. One class of these assists in evaluating color choices (e.g., Ou et al. [38] created a prototype for evaluating color schemes on the basis of color semantics and harmony models). Another category helps designers avoid color mismatch by constraining the colors available for selection. An example is Hu et al.'s [13] color-palette authoring tool, which ensures that all colors used share at least one property ("familial factor") and that the spread for each other property shows equal distances ("rhythm span"). Another constraint-based tool is ACE [46], which helps guarantee only accessible color combinations.

Several assistance tools visualize color schemes in contexts other than palettes. Exploring new forms of interaction with color pickers, Shugrina et al. [44] introduced a color-scheme authoring tool that

[2]See `https://color.adobe.com/create/color-wheel` and `https://paletton.com/`, both as accessed on October 13, 2022.

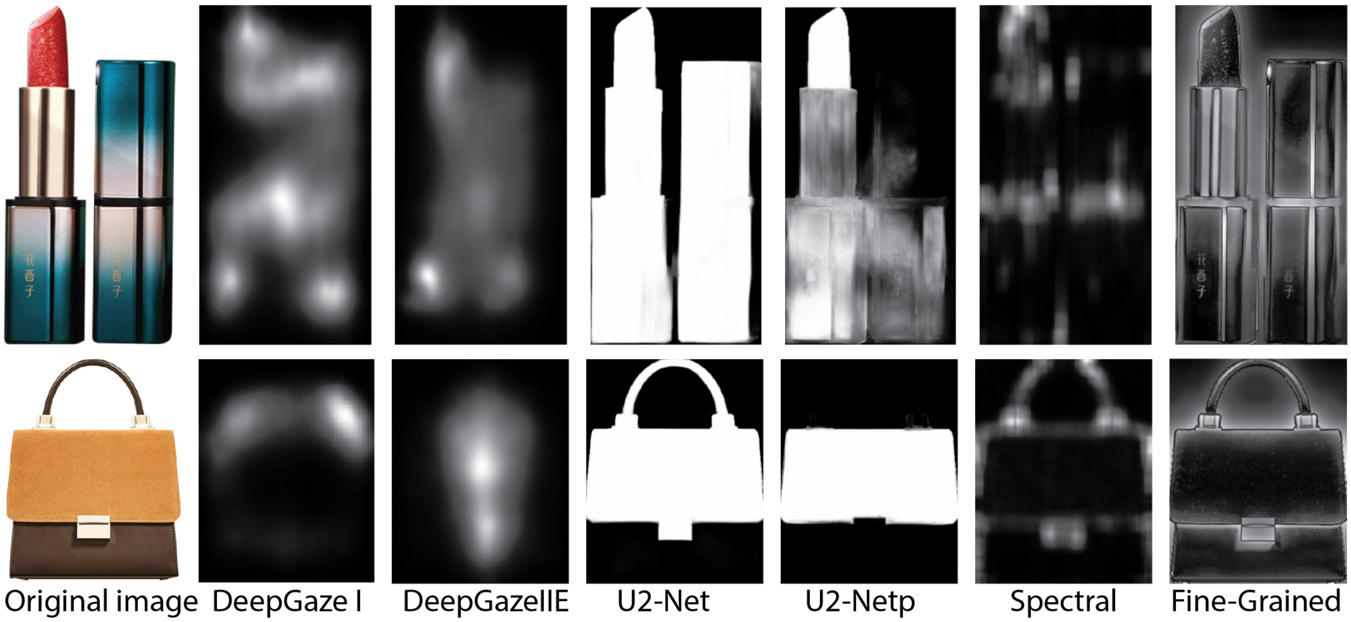

Figure 2: An illustration from the range of saliency models. DeepGaze I and IIE predict fixation location, U2-Net and U2-Netp predict the region of the mail object, Spectral and Fine-Grained predict salient regions based on pixel-level features.

supports more versatile interactions with colors by considering gradients of various sorts and presenting colors in the context of a vector graphic. Meier et al [30] developed a set of extensions to Adobe Illustrator that allow one to browse palettes, adapt and rearrange them, consult reference images, preview how one's color choices look with geometric shapes, and even browse shades by name to uncover a "brown" or "sand" which are unintuitive to find in a standard color picker. Devised in light of interviews with designers, artists, scientists, and engineers, color portraits [20] informed four interactive tools, for 1) interacting with palettes, which are visualized in compositions of rectangles; 2) combining colors with texture; 3) linking them with time (the history of color options); and 4) using color to reveal a project's work process. Whereas these tools illustrate the color scheme in graphics other than the final design, Playful Palette [43] and Color Triad [42] enable color-mixing interactions and recoloring of existing artwork. Also, the system of Mellado et al.[31] lets one explore palettes for a given image by setting up a graph structure; maps between colors define constraints, which enable simper switching of colors, updates to other colors, and interpolation between palettes.

An alternative approach is to generate and suggest design ideas. A tool for exploring UI layouts via sketching, Sketchplore [47] considers color so as to avoid visual clutter, facilitate visual search, and offer color-harmony-based recoloring recommendations in line with color-wheel templates. Meanwhile, Phan et al. [39] developed a sorting and clustering algorithm for palettes and considered aligning suggested palettes with the user's work.

The emphasis across the landscape of color-scheme tools is confined to picking colors based on color theory or from art. Authoring tools that consider the graphical design at hand while also supporting the full workflow of choosing and applying colors have remained absent.

2.5 Automated Coloring

Prior work has led to systems that can automate coloring of graphical designs. Vinci [9] automatically generates poster designs when given only the main image and text. It decides on placement, colors, and embellishing of graphical elements and creates compelling designs that fall short of human performance in relatively few respects.

Furthermore, there are systems that account for semantics and legibility requirements when automatically coloring magazine covers [19] and web designs [8]. While these automate color design, with only minimal designer inputs, we pursued a technique wherein designers control the extent of their influence over the final result and the steps for reaching it.

2.6 Summary

Extracting colors from images frequently serves the aim of representing the colors via a smaller palette. This process typically considers the whole image. Approaches that prioritize salient areas show potential, however. A broad range of models – both theory-based and empirical – exists for finding palettes with well-matched colors, and techniques assign and apply colors from a palette to the design elements in special cases such as patterns, websites, and grayscale images. Dedicated approaches exist also to automate the coloring of posters, websites, and magazine covers. Notwithstanding these advances, no interaction techniques have yet been proposed that combine these approaches to enable interactive exploration of the full coloring workflow for a given graphical design.

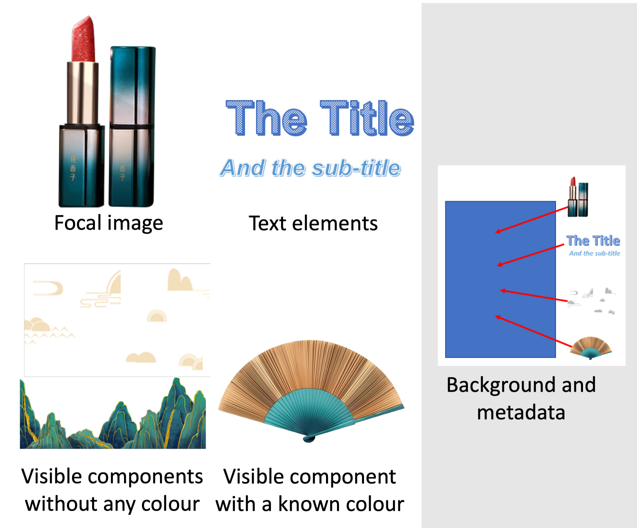

Figure 3: Components of one specific scenario for a typical coloring-design task.

3 AN INTERACTION WALKTHROUGH

The design of the interaction between the designer and the CoColor-implementing Figma plugin constitutes a vital part of the contribution. We present the overall flow first.

Work with CoColor begins with the designer choosing a focal object, such as an image. This is the content around which the color design is to be built; e.g., the focal image might be a photograph/illustration of the product to be marketed via the graphical design. In addition to the canvas specifications and the focal object, the interface expects further elements on the canvas, in the form of *visible components* (title, subtitle, decorations, corporate logo, etc.) and *metadata* (recommended locations, sizes, and predefined colors). The product image, visible components, and metadata collectively constitute the specified scenario for a typical coloring-design task, as illustrated in Figure 3.

Step 1: Extract Base Palettes:

Taking the focal image as the starting point for defining the initial color scheme guarantees its harmonious embedding in the overall color design. Our technique utilizes diverse approaches (described further along) to identify the key colors within this image. For example, one might pick the colors used most within those of the image's elements identified as salient. Another option is to identify and remove the background, then pick the colors used most in the foreground elements. Alternatively, statistical sampling of the colors from all pixels enables identifying the median color clusters in the image. These approaches produce distinct color schemes, from which the designer may choose. We set the default number of extracted colors to 3, as shown in Figure 4's examples of extraction from the foreground of several focal images. This is enough to represent most types of color harmony [17] without risking a cluttered look from adding more colors (though chosen with care, this value for the parameter can be changed easily).

Figure 4: Step 1 – automatic extraction of key colors from a given focal image.

Step 2: Extend the Palettes: For a well-rounded palette with colors that suit the variety of elements in a typical design, additional colors are needed. Generating these from a source different from the focal image provides the contrast necessary to accentuate the image. Feeding the existing colors to a color-harmony model ensures that the extension matches the image and the base palette. Figure 5 presents a typical result from our technique's palette-extension stage. Although the figure shows only the top recommendation, our technique offers a wide range of options from which to select new colors. By default, CoColor extends the palette by one color, but more may be added. Designers can use this feature to explore the palette space beyond the focal image by requesting an extension to user-defined palettes.

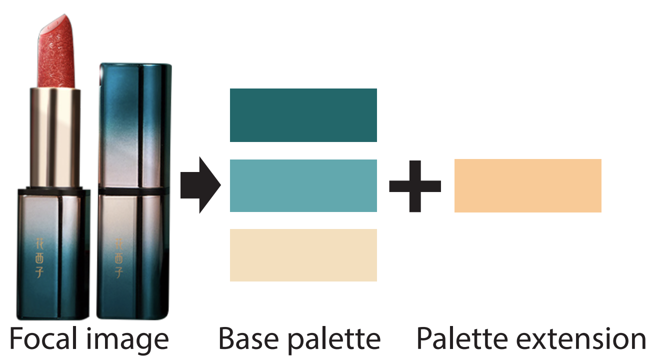

Figure 5: Step 2 – extension of the palette. Given a base palette of colors extracted from the focal image, CoColor adds one or more additional colors.

Step 3: Apply Colors to Graphical Elements: After the palette extension, there are enough unique colors to cover all elements encompassed by the design task. Specific colors now get assigned to specific elements, on the basis of several factors: CoColor accounts for foreground elements' visibility against the background, in conjunction with applying a novel assignment heuristic to balance the physical distance between every two visible elements with the logical separation of the colors assigned to those elements. The technique also considers the visual saliency of the focal image in the resulting design. Varying the various governing factors' relative weights yields a rich array of designs from a single color palette. Figure 6 offers an example; all these posters were created from one palette.

Figure 6: Step 3 – apply colors to the design. Here, the designer is offered three distinct options, based on a single color schema.

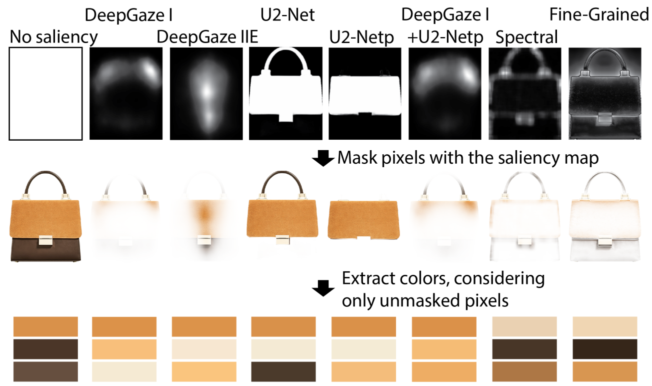

Figure 8: By means of multiple saliency models, several saliency maps are created that highlight the foreground or the most eye-catching parts of it. These maps are combined with the image before extraction of colors. This leads to a variety of color palettes.

Should the fully colored designs available at this stage provide insufficient contrast for accessibility, Web Content Accessibility Guidelines (WCAG) relative-luminance calculations can identify this issue. If the colors of a specific pair of connected elements do not support legibility, one or both of those colors must be changed. After the color-change requirements' collation throughout the design, the smallest possible color changes are made such that legibility is assured.

CoColor supports multiple coloring choices to be simultaneously designed, and offered for inspection and iteration within the design task's setting. Figure 7 offers an example of the results: potential options generated for a single design scenario. An evaluation function rates the results and sorts them for the designer accordingly.

Figure 7: Expected outputs – multiple well-rendered posters.

4 THE COMPUTATIONAL APPROACH

This section explains the computation methods applied in the three steps characterized above. Besides supporting rapid exploration, our technique pursues three objectives articulated for ensuring the quality of suggestions made to the user:

(1) *Saliency of the focal image*: Guided by designers' own ways of highlighting images with color [22], CoColor obtains the main colors from within the focal image and drives the overall color scheme on their basis. These colors, when used in other parts of the design, can both refer to the focal image and help synthesize a seamless whole. We strove to pick the colors from salient portions of the image, to model designer preference for those colors [22, 27].

(2) *A balanced overall color scheme*: While colors from the focal image connect it with the design, contrasting colors provide a necessary means of highlighting it as important [32]. Hence, the theme is enriched with colors from a separate source. For overall coordination and the colors' applicability, a palette model trained on a large dataset of real-world designs evaluates the scheme generated.

(3) *Contrast of text elements*: The metadata identify the text components. Color contrasts between textual elements and their backgrounds can be deduced in conjunction with the other decisions regarding color palettes for visual components. Hence, the text elements' legibility (in WCAG terms [3]) in their context can be verified.

(4) *Provision of options*: The optimal color selection from the focal image, palette extension, and color assignment depend on context and designer preference. Therefore, our technique employs several methods, tailored to different scenarios. The designer can choose from among multiple design options or browse them to seek inspiration for further iterations.

A schematic overview of the pipeline is given in Figure 9.

4.1 Step 1: Extract Palettes

Since the final result should embody harmony that fits and supports the focal object, our algorithm takes that object as its source for the base palette. The process behind the base-palette generation is depicted in Figure 8. We apply eight distinct techniques for the color extraction, seven of which utilize prior machine-learning approaches or theory-based models to detect salient areas of the focal image. They identify the most prominent parts of the focal

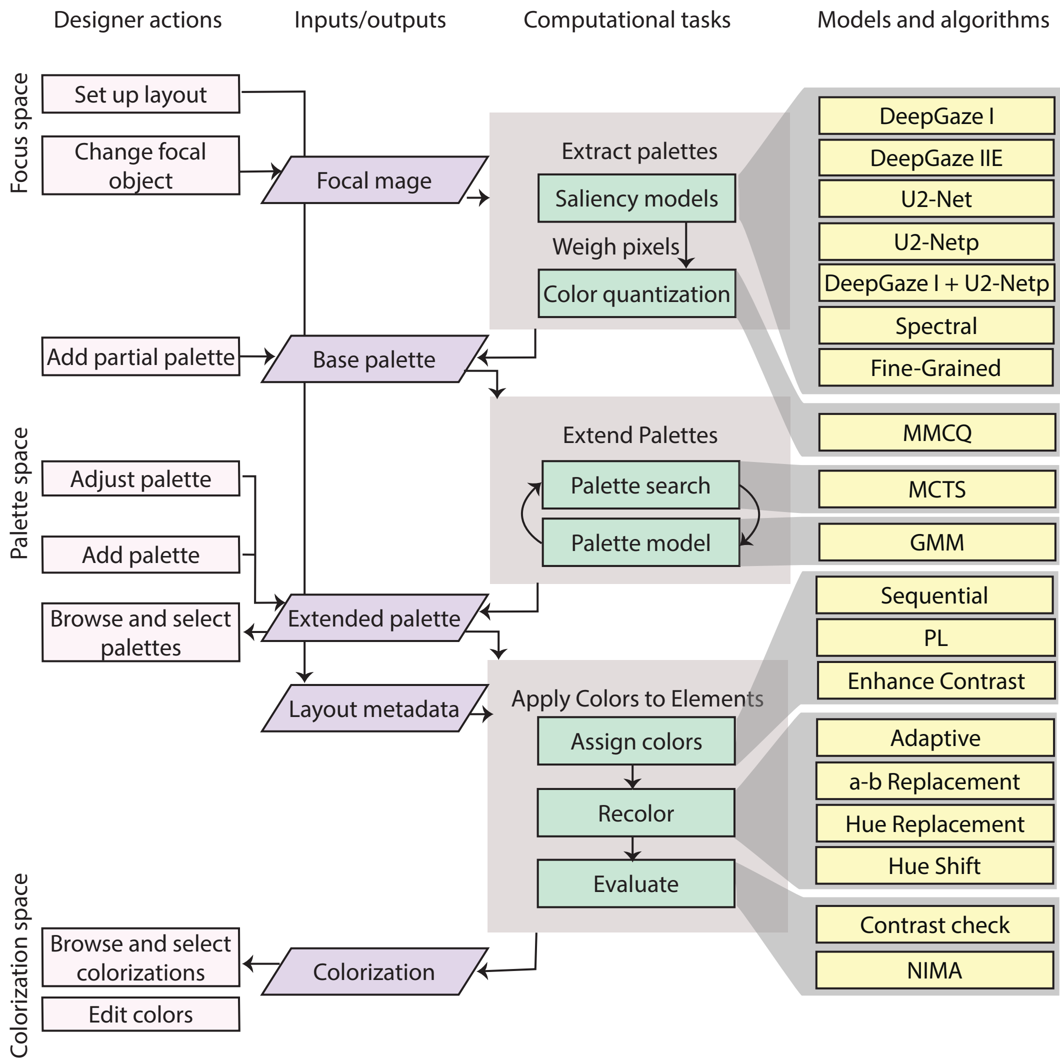

Figure 9: Computational approaches employed to support exploration of color designs in CoColor and how they link to the computational tasks, and designer actions.

object – those that should catch capture human attention most readily – and picks colors from these areas only.

The saliency maps from the individual models complement each other. The U2-Net models [40] are designed to distinguish the main object from the rest of the image. These models work well if the focus within the image should be on this object, as in the case of a typical product image. In contrast, the DeepGaze models [26, 29] predict salient pixels beyond the main object too. This is advantageous if the background contains supportive salient colors or when only some parts of the main object are relevant. To address the latter case, we implemented an additional model also. It runs U2-Netp on the image to identify the main object, then DeepGaze I to detect salient parts of the map. The models denoted as Fine-Grained [12] and Spectral [33] detect areas of high contrast / high visual information. The resulting contrasting palettes can lead to visually interesting designs. Also, Spectral's short run times make this model's results available to users sooner.

The maps from the saliency models predict the saliency of every pixel as a value from 0 to 1. Before extracting colors via modified median cut quantization (MMCQ) [1], our method weighs the pixels, using the values from the source saliency map. CoColor performs color extraction once with each saliency map, thus generating seven base palettes. An eighth palette is created by MMCQ applied over the full image.

4.2 Step 2: Extend Palettes

Once focal colors are pinpointed, the method adds colors to the base palette with the aid of Monte Carlo Tree Search (MCTS) [5] and a probability distribution over the space of the palettes, per Figure 11. For the probability function, we extracted four colors

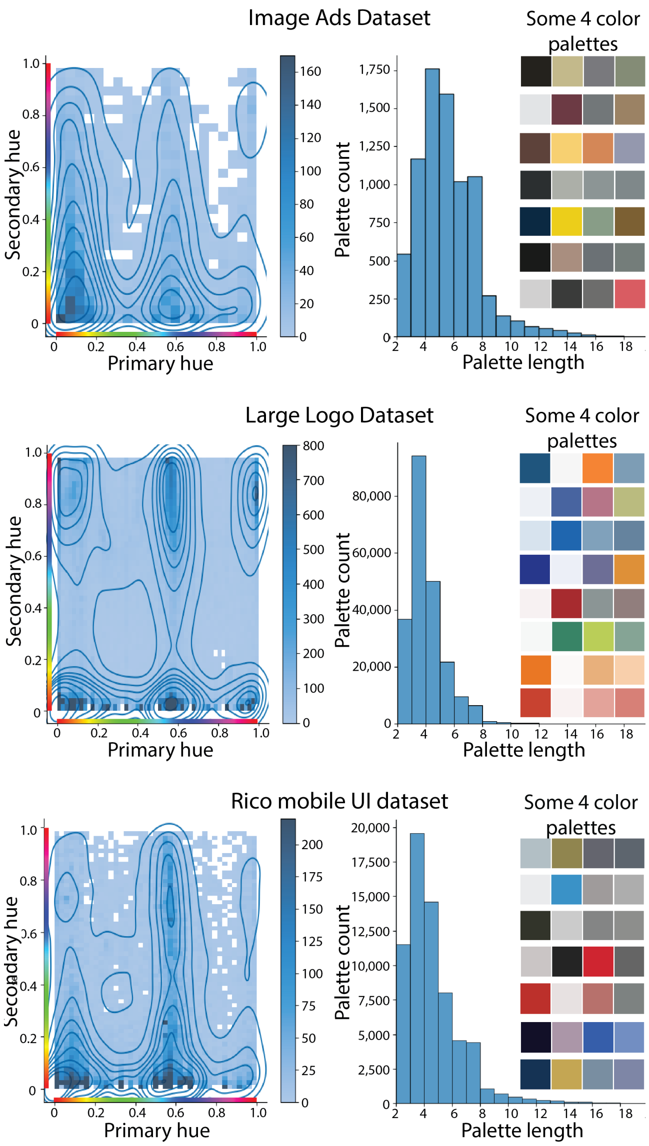

Figure 10: On the left, histograms and distributions of the hue combinations in the palettes from each dataset, where the primary hue represent the biggest cluster center while the secondary one represents the second-biggest. At the right, histograms of the number of colors (palette length) extracted from the images in the dataset (the optimal number was ascertained via the *Elbow Method*). Next to the histogram are several representative four-color palettes, to illustrate the colors in the dataset.

extracted from the pixels of three datasets composed of graphical designs: 221,369 sharp logos and icons from the Large Logo Dataset (LLD) [41], 7,885 images from the Pitt Image Ads Dataset (PIAD) [16], and 66,246 screenshots of mobile user interfaces from the Rico dataset [7]. We fitted a Gaussian mixture model (GMM) with 16 components to the 12-dimensional palette space (4 colors × values for hue, lightness, and chroma) for each of the datasets. The MCTS explores the palette space while keeping the colors of the base palette constant, and it returns palettes that are among the 20% most likely to appear in the dataset.

To decide on the default dataset, we analyzed the quantities of colors in the images. This entailed clustering the colors via k-means with increasing k and computing the compactness (squared distances to the cluster centers). Via the Elbow Method, we ascertained the point where compactness stops decreasing significantly, thus finding the optimal number of colors to extract for each image. From the histograms of the palettes' hue combinations and palette sizes, presented in Figure 10, the mean palette size was 3.58 for LLD data, 5.15 for the PIAD material, and 4.16 for the Rico dataset. In all three cases, the hue-combination histogram reveals that hues close to red (inclusive of completely desaturated colors, with a hue value of 0) show the highest density. High density is visible also for cyans and blues, along with combinations of warm colors with cyan/blue and of cyan/blue with warm colors. Although these results are consistent with findings from prior work on color preferences [15], the latter correlation was not observed in the PIAD palettes. The Rico dataset shows an additional pattern wherein cyans/blues are frequently combined with any hue, while the LLD shows the expected dip around greens [15]. Example palettes from each dataset attest that those from the LLD tend to contain more pleasant colors and combinations thereof, possibly because shadows, decoration, and background graphics are less prevalent in these images. On account of these properties, we decided on LLD as the default dataset in our implementation.

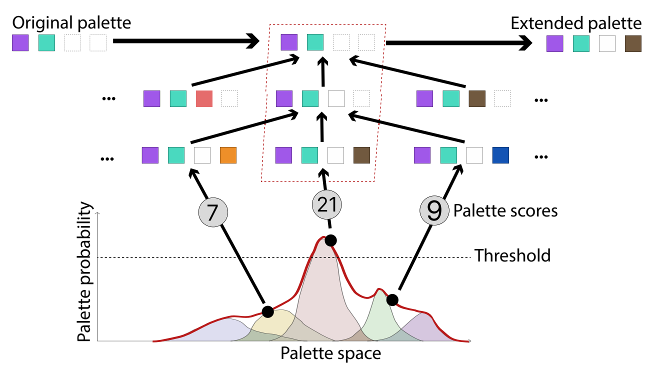

Figure 11: Using Monte Carlo Tree Search iteratively with a Gaussian mixed model to generate complete palettes, given an incomplete color palette. Palette at the leafs are scored based on their probability in the palette space. The candidates crossing the threshold are considered the best choices.

4.3 Step 3: Apply Colors to Elements

Generation of the final palette is followed by assigning its colors to the elements composing the design, with attention to aesthetics and function. We developed three assignment algorithms in response to this challenge, called Sequential, PL, and Enhance Contrast (see Figure 12).

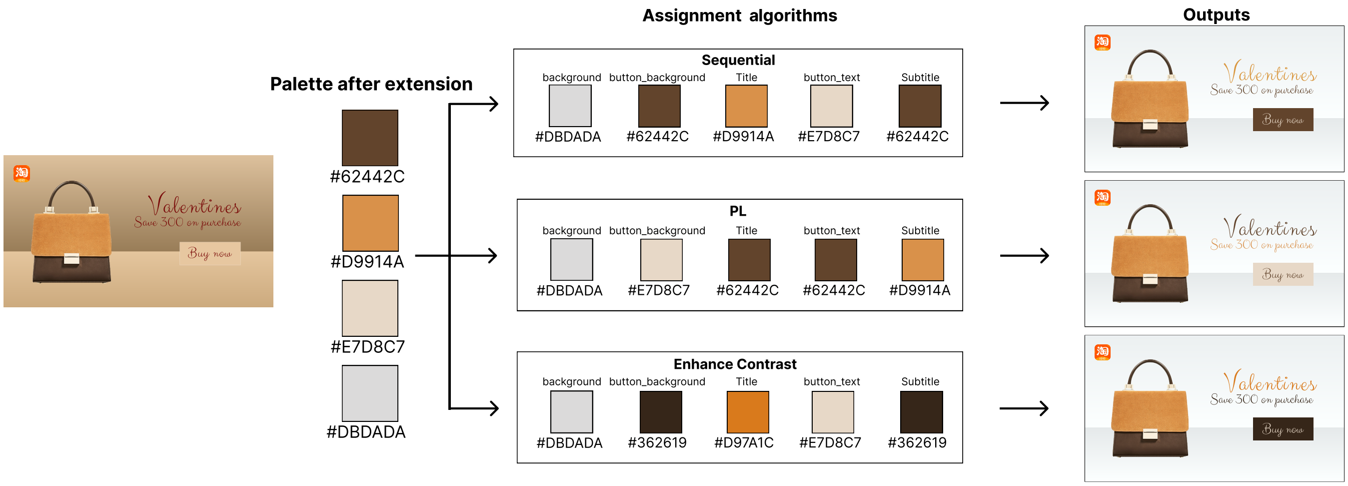

Figure 12: Assigning colors from the selected palette to the elements of the design by means of the three assignment algorithm – based on sequence, physical logical distance, and enhancement of contrast.

The *Sequential* algorithm assigns the colors to the design's layers sequentially, starting by using the extended color for the background. Since this is the only color not from the product image, it guarantees at least some contrast for all the foreground elements. Next, the distances between the base palette's colors and the extended color (i.e., the background color) are calculated (root mean squared (RMS) of `color_1` and `color_2`) and sorted in descending order. These colors then get assigned to the list of the original design's non-background layers. The key assumption underlying this approach is that upper layers are most likely foreground elements so should stand out in the sharpest contrast to the background.

Our second algorithm, *PL* (for physical logical distance), assumes that foreground elements are non-overlapping and orders elements by their location in the layout rather than their position in layer order. It uses the center of each element to calculate the distance to the focal image (*RMS*(`focal_object, layer`)). The algorithm also finds the distance between the colors in the extended palette and the average color of the focal image. These distances are sorted in descending order, and then colors are assigned sequentially on each order's basis.

The third mechanism, *Enhance Contrast*, combines parts of the Sequential approach with a contrast check and adjustment. Firstly, it evaluates the average color of the base palette. Next, it sorts the colors by their distance from the average color, in descending order. The color most distant from the others goes to the background, and the remaining elements are assigned colors in sequence from the lower layers to the topmost. In this process, the algorithm checks the relative lightness of overlapping elements. If it finds that contrast to be below a given threshold, the lightness of the element in the higher layer is adjusted in CIELAB, a perceptually uniform color space with a separate lightness channel.

4.4 Recoloring

Within each layer of the design, the original color for every element must be replaced by the newly assigned one, but simply overwriting the old color does not work – we could lose important details such as gradients, borders, textures, and patterns. To preserve these details, we implemented four methods, hereinafter referred to as the Adaptive method, A–B Replacement, Hue Replacement, and the Hue Shift method. Figure 13 compares these methods' performance for a gradient image. For all of these methods, the colors are translated into either the CIELAB or HSV color space before computing of the new pixel colors. Using these color spaces exploits their separation of the hue and lightness information into distinct channels: CIELAB has the aforementioned lightness channel plus two channels that represent the hue (a* and b*), and HSV has one channel each for hue (H), saturation (S), and value or brightness (V).

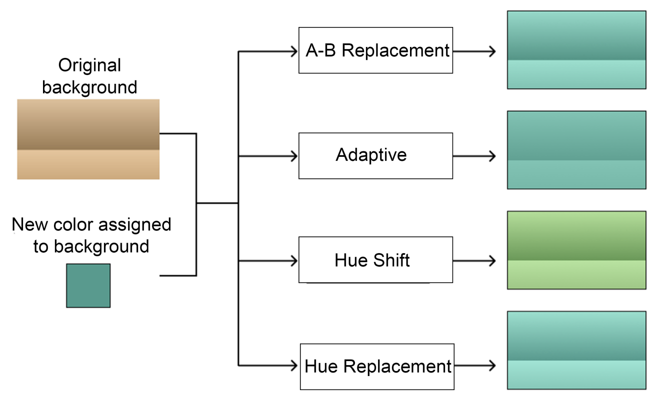

Figure 13: Using the assigned colors to recolor the original element in the design by means of four recoloring algorithm.

The first step of the *Adaptive* method is color quantization on the layer. The colors in the resulting palette are compared to the assigned one (delta in the CIELAB color space). If the results all are below a threshold, thus suggesting the colors' similarity, recoloring proceeds by averaging each pixel color with the assigned color (in the RGB color space). If the difference is above the threshold, the color space is converted to CIELAB, and the a* and b* values of the old color are replaced by the new ones. The L* value is shifted by the difference between the dominant and the assigned color.

In the *A–B Replacement* method, the a* and b* values of the original color are replaced by the newly assigned color. The L* value, though, is an average between the original and the new.

Similarly, the *Hue Replacement* method replaces the relevant hue with that of the new color, in terms of the HSV color space, and the saturation and the value are averages from the old and the new color.

Finally, the *Hue Shift* method too averages the saturation and value of the old and the newly assigned color; however, instead of replacing the hue, the algorithm shifts it by the difference between the assigned color and the most dominant color – i.e., the color representing the most pixels after color quantization.

4.5 Evaluation and Refinement

One additional step of the workflow is the *repair* to enhance readability or legibility. This takes place after the recoloring step of the pipeline as an additional measure to make sure results are satisfactory. The algorithm compares the colors between overlapping layers, and if the contrast is low according to WCAG standards, it shifts the color's luminance to increase it.

The algorithm's final step is to evaluate the results such that they serve sorting and filtering of designs. For this, we employed the NIMA model [45], which captures aesthetic perception of images by means of a convolutional neural network architecture. Its scores for images are reliable, demonstrating strong correlation to human perception. This model differs from other scoring models in that it outputs a distribution of human opinion scores. It was trained on a large body of material: the AVA dataset [35], which contains 255,000 images rated for their aesthetic qualities by photographers. CoColor uses the pre-trained model to assess the aesthetics of the generated designs quantitatively.

4.6 The Figma Plugin

Figma is a design tool in widespread use by designers of user interfaces for several forms of graphical design. We built our plugin on top of Figma's existing plugin system, which is written in the language TypeScript. After loading, the Figma plugin prompts the user to supply a frame (a pre-designed layout to be colorized). When the frame has been retrieved, the plugin detects the focal object (as Figure 1.a shows), by finding the largest non-background image within the frame. By clicking on the "Change anchor" button, the user can choose another layer as the focal object or decide which layers will be colorized in the colorization step. The plugin then begins generating a list of palettes and a set of colorized results as described above. Users can filter out results on the basis of the palettes involved or generate results based on their custom palettes. Clicking on any result replicates it as a new editable frame on Figma's document canvas. This allows users to engage with the process as a unified iterative design and colorization entity. The plugin converts Figma frames into data instances of the form expected by the back end and supports both vector and raster layers.

ID	Design studies (years)	Work experience (years)	Fields of design
P1	2	3	UI, UX
P2	6	3	Product, graphic, service, strategy
P3	1	4	UX, UI
P4	7	3	Service, UX, product
P5	4	3	UX, UI
P6	2	0	Service, web, UX, strategy
P7	6	17	Graphic, UI, UX, service
P8	7	2	Graphic, 3D
P9	1	0	Design
P10	6	6	Graphic, UI, UX, motion, service
P11	1	0	UI, UX, service
P12	1	1	Graphic, UI, UX, concept
P13	2	1	Service, UI, interaction
P14	1	7	Graphic
P15	3	1	UX, UI, service
P16	2	0	UX, UI, digital drawing

Table 1: The user-study participants and their level of university-level education in a design-related field, years of work experience as a designer, and design specialty.

5 DESIGNER STUDY

We conducted a user study to evaluate whether and how our interaction technique assists in color design. With aims of evaluating the technique's support for designer creativity, ability to explore different outcomes, and the subjective experience, we formulated the following goals to guide the design of the study:

(1) Understand whether the technique allows designers to generate feasible designs more easily and quickly
(2) Ascertain how well the interaction technique supports the designers' creativity
(3) Understand whether the technique supports exploration better than a no-support baseline
(4) Assess whether the final colored designs are higher-quality and more satisfying when developed with CoColor
(5) Understand how designers utilize each of the three steps in CoColor
(6) Find out whether designers deem CoColor helpful for their practice

Accordingly, in the study design we aimed for realistic tasks, a representative sample of designers, well-validated standard metrics coupled with probing the participants' subjective views of their experience, comparison against a baseline, and sufficient familiarization time to 1) enable learning the functions and avoid later learning effects and 2) allow participants to develop a workflow with the plugin.

5.1 Participants

We recruited 16 people with at least one full year of work experience or design-related studies. Email lists of local designer associations and groups were used to reach out to potential participants. They were required to have chosen and applied colors to at least one design in the last 12 months and possess experience with graphic- or UI-design software. We enforced these requirements with a screening questionnaire before the start of the study. Participants' ages ranged from 22 to 40, with a mean age of 28.4. Ten were female, five were male, and one preferred not to indicate a gender. Table 1 characterizes the design specialties, and experience levels. Before taking part in the study, all subjects were informed of the study's conditions and agreed to them. They were compensated for their time with 30-euro vouchers for a local restaurant.

5.2 Materials

Each participant was seated at a desk with a laptop computer and an additional screen. They used the laptop for questionnaire completion and watching a tutorial video about the functions of the plugin, while the general Figma interface and our plugin were ready on the other screen. Three Figma files had been prepared. One contained a practice task for familiarization with Figma, a task for practice with the plugin, and a sample realistic case for developing a workflow with the plugin. The other two for the experiment proper, contained realistic design cases. Each of these design tasks comprised four pages, with the cases consisting of a brief on the client and on the design's purpose and a layout that needed to be colored. The layouts for all cases were inspired by various designs collected online but contained only images under open licenses for non-commercial use. All colors except those in images had been replaced with shades of gray, and the layouts were adjusted such that all briefs and all cases' types and quantities of layers exhibited comparable complexity. The designs spanned different media and topics. We used standard questionnaires to assess the task load of each case (NASA-TLX [11]), the creativity support (CSI [4]), and usability (SUS [2]) of the plugin and the baseline setting. The team used custom questionnaires to collect subjective ratings for the resulting color designs and evaluate the popularity of individual plugin features. Furthermore, there was a final interview, using a three-question script: Firstly, we asked the participant to compare the coloring task with vs. without the plugin. The second question which condition involved easier exploration of the various options and why. Finally, we asked which features of the plugin would mesh well with their practice and which would not.

5.3 Experiment Design

A within-subject design exposed participants to two conditions. In one condition, participants colored two or more cases from one set by using the plugin freely in combination with Figma's standard features. In the other condition, the baseline, they colored two or more cases from another set, using only Figma features. The order of the design sets and that of the conditions both were completely counterbalanced; i.e., there were four groups. The dependent variables were the questionnaire responses, tool-interaction data, and design outcomes.

The collected data related to the goals of our study (see section 5 as follows.

(1) Ease and efficiency: NASA TLX, amount of designs finished per condition, interaction data, SUS
(2) Creativity support: CSI, final interview
(3) Exploration support: CSI, final interview
(4) Quality and satisfaction with results: Custom questionnaire (after every task), design outcomes
(5) Utilization of the supported three steps: Interaction data, final interview
(6) Helpfulness in practice: SUS, custom questionnaire (after plugin condition), final interview

5.4 Procedure

After reading and signing the consent form, the participant was given our questionnaire form addressing design background. This was followed by a tutorial introducing the relevant functions of Figma, including practicing with them, to ensure a certain level of Figma skills. Next, a video explained the layout of the plugin and how to operate its functions. After watching the video, the participant could try out all functions, following an additional step-by-step tutorial. This initial portion of the study took approximately 20 minutes in all.

Next was the main part of the study: the task of creating three compelling and functional color-design versions for each case that could serve as a suitable basis for discussing color options with the hypothetical client. Participants were allowed to edit the position, size, and font of elements in the layout but were encouraged to focus on the colors. For every case, the participant was given 10 minutes to create the set of color designs. The research team treated the first case with the plugin as a practice one, to reduce any effect of learning during the recorded cases. In each condition, participants performed the task for at least two cases. The other two cases in each set were available in case the designer worked quickly enough to handle more tasks in the time allotted. After each case, the experimenter asked the participants why they chose these particular three versions to show to the client and, if this information was not already provided, solicited details of what was deemed good about each version. Then, the participant rated the designs created and the task load. The between-task questionnaire and interview stage lasted approximately 5–10 minutes. After each condition, participants had about five minutes to rate the usability of the overall system and its creativity support. After the plugin condition, participants also rated the usefulness of the plugin features. When all tasks were complete, the experimenter conducted the structured interview, which took roughly 10–15 minutes. The interviews were voice-recorded transcribed by one of the researchers. Also, all onscreen interaction and audio during the tasks were recorded. Additional data were furnished by logging of interactions with the plugin and color edits in Figma. The full procedure took approximately two hours for each participant.

5.5 Data Analysis

We computed unweighted NASA-TLX scores, in line with common practice for ensuring more robust results [10]. For CSI scores, we followed the approach described by Carroll et al. [4], and SUS scores too were computed per established practice [2]. To test for statistical significance, we performed signed-rank Wilcoxon tests suitable for comparing within-subject samples on all scores and ratings.

We counted interactions with the plugin and all color edits outside it. Furthermore, we analyzed how often each of the individual models and algorithms from our implementation contributed to the palettes/colorizations selected by the designers.

Qualitative data (from the design outcomes, interview transcripts, etc.) were analyzed for recurring themes and common features. Two of the authors independently performed thematic analysis and developed a view of shared-meaning themes from the data. On this basis, the authors agreed on one combined set of themes.

6 RESULTS

6.1 Quantitative Findings

6.1.1 Interaction. On average, participants in the plugin condition interacted with the palettes and colorizations in it 25.1 times per case. The most frequent interaction with the plugin was filtering by palettes (with 15.8 instances, on average). Every designer used this feature. All participants but one interacted with the plugin at palette-creation level. Most participants (11 out of 16) added custom palettes, and nine altered the palettes in the palette list. Participants selected, on average, 4.4 colorizations per case (range: 1–14). Of the frames propagated to the final set of designs, those created without the plugin had been edited for their colors 13.1 times, on average, and the designers edited the ones in the plugin condition only 3.1 times post-selection.

We found that 306 of the palettes clicked (9.56/task) were extracted from images after application of a saliency map and 59 (1.84/task) considering all pixels of the image equally. Designers also frequently filtered by palettes they had edited or added (140 in all, or 4.38/task). A slightly higher proportion of designer-defined palettes was visible in the colorizations selected (see Figure 14). When looking at the click counts associated with each saliency model (presented in Figure 15), we could see that the theory-based spectral model resulted in the most popular palettes. Interestingly, the model combining DeepGaze I and U2-Netp led to the least popular ones even though this model was the best at detecting only salient areas. Division of clicks across similar models might account for some of these patterns, though: the spectral model may have benefited from being the most distinct from the others, whereas similar models (the two DeepGaze ones and the combined one, or the two U2-Net models) might have competed for clicks. When considered jointly, similar models attracted more clicks. In addition, Spectral may have gained clicks from being swifter than others: with the shortest computation time, its palette was often available before the others. As for the clicks on colorizations, examining them by color-assignment method revealed that the Sequential algorithm, while yielding the most popular colorizations, showed only small differences from other methods, as is visible at the left in Figure 16. The click counts for the various recoloring methods display larger differences in clicks: the adaptive method occasioned 2.75 clicks per case, followed by A–B Replacement at 2.09 clicks, then Hue Replacement (with 1.69) and finally Hue Shift (at 1.19) (see the right-hand portion of Figure 16).

6.1.2 NASA-TLX. There was no significant difference between conditions in the overall NASA-TLX scores. This instrument's individual scales showed a significant difference only for physical load (Z=15, p=0.009); the average physical burden was slightly smaller with the plugin. Still, the value was low in both conditions (20 with the plugin, 27.2 without). Most of the physical load was due to operating the mouse. The amount of clicking was noticeably lower in the plugin condition, likely explaining the difference (see section 6.1.1). Figure 19 shows the frequency for each of the individual NASA-TLX scales.

NASA-TLX	Z=208	p=0.61
Mental demand	Z=160.0	p=0.94
Physical demand	**Z=15.0**	**p=0.009**
Temporal demand	Z=184.5	p=0.67
Performance	Z=160.5	p=0.7
Effort	Z=163.5	p=0.36
Frustration	Z=184	p=0.66
Self-reported evaluation		
Option A	Z=192	p=0.8
Option B	Z=189.5	p=0.76
Option C	Z=124.5	p=0.3
Basis for discussion	Z=122	p=0.88
Selection for choice	Z=168.0	p=0.61
CSI score	Z=46.0	p=0.68
Enjoyment	Z=48	p=0.49
Exploration	Z=40	p=0.25
Value for the Effort	Z=46.5	p=0.7
Expressiveness	Z=58.5	p=0.67
Immersion	Z=30	p=0.16
SUS score	Z=58.5	p=0.93
SUS Q1	Z=21.5	p=0.3
SUS Q2	Z=37	p=0.87
SUS Q3	Z=21	p=0.86
SUS Q4	**Z=0**	**p=0.04**
SUS Q5	Z=13.5	p=0.07
SUS Q6	**Z=4**	**p=0.008**
SUS Q7	**Z=0**	**p=0.01**
SUS Q8	Z=10	p=0.86
SUS Q9	Z=20.5	p=0.86
SUS Q10	Z=8	p=0.13

Table 2: Z-statistics and p-values from signed-rank Wilcoxon tests.

6.1.3 Creativity Support Index. Neither the overall Creativity Support Index instrument nor the individual components showed a significant difference in the two conditions. The mean CSI score was 69.3 with the plugin and 67.2 without. Figure 20 presents a breakdown of the responses.

6.1.4 System Usability Score. Overall SUS ratings displayed no statistically significant inter-condition differences. However, for three of the statements individually, we did find significant differences in agreement ratings: Use of the plugin correlated with slightly less agreement that the user would need technical personnel's assistance before being able to use the interface (SUS item 4: Z=0, p=0.04), though the median was 1 in both conditions. With the plugin, participants agreed more with the item-6 claim that too much inconsistency was present (median: 2 with plugin, 1 without plugin; Z=4, p=0.008), but in the plugin condition they agreed more strongly that they would imagine most people learning to use the

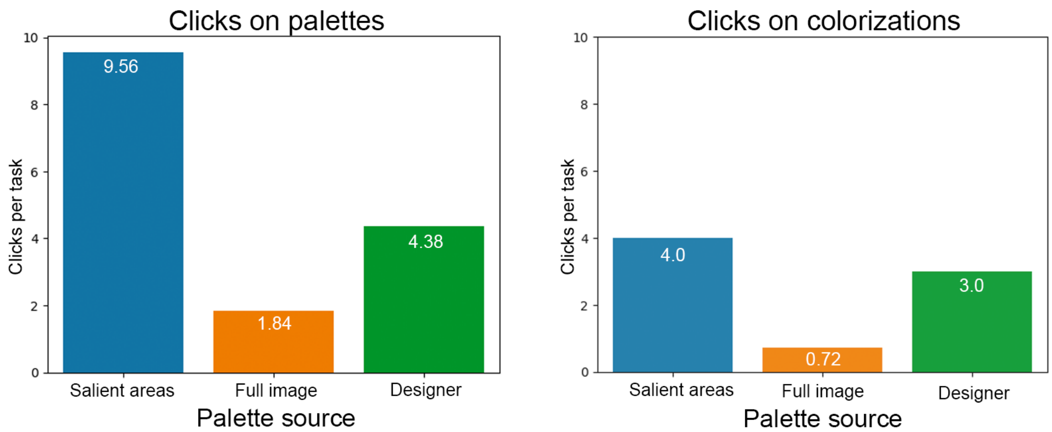

Figure 14: The palettes’ popularity among study participants, as judged by clicks – average clicks per task on the palettes in Figure 1’s pane g (at left) and on the colorizations in its pane i (at right). The most-clicked palettes were extracted from the salient areas of the focal image with participants clicking on their own palettes and the resulting colorizations second most commonly.

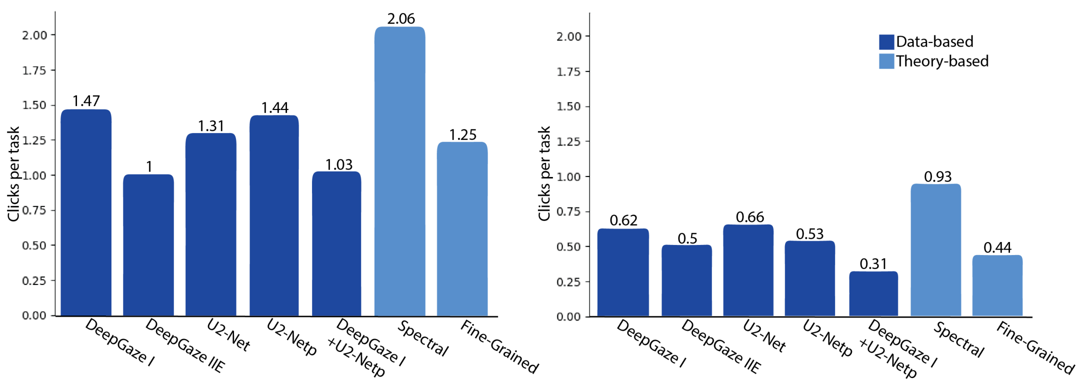

Figure 15: Average numbers of clicks per task plotted against the type of saliency model. At the left are clicks on palettes to filter in colorizations applying said palette. On the right are clicks on colorizations.

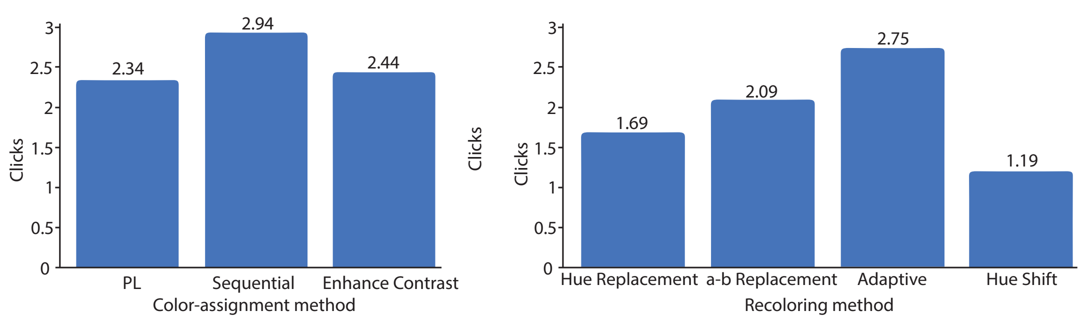

Figure 16: Colorizations’ average number of clicks per task, by the color-assignment (left) and recoloring-method (right) algorithms used to create them.

interface quickly (median: 4.5 with plugin, 3.5 without; Z=0, p=0.01). Figure 21 itemizes the responses by SUS questionnaire item.

6.1.5 Rating of Plugin Features. The plugin’s individual features received favorable ratings. Scores for its options for choosing which layers to recolor, browsing colorization options, and adding palettes were the most positive, with a median score of 7/7 for all these.

Selecting a focal object, filtering, and refining palettes were rated 6/7. Figure 22 presents the feature-specific utility ratings in full.

6.1.6 Performance. No significant differences between conditions were evident in the designers' evaluations of either their individual coloring results or the set to choose from or base a discussion on (see Figure 23 for the self-reporting details). We also considered the amount of time needed for completing the task; most designers used all the time in both conditions. Table 2 provides details of the statistical tests.

6.2 Qualitative Findings

Participants gave positive feedback overall, especially on the opportunity for exploring the coloring options. Some positive comments about usefulness were of a general nature, as with P5's note that "everything is there nicely" or P6's "pretty intuitive." Other designers were more selective in their praise, liked only some of the features, or had ideas for improvements. We give details of the qualitative feedback below, applying boldface to the themes that emerged.

6.2.1 Exploration. Of the 16 participants, 13 clearly stated that the plugin provided **ease of exploration** , with P11 describing it as "great" for exploring options quickly and P13 commending the reduced clicking and mental effort. Likewise, P14 praised the expedited exploration and called the plugin "a great shortcut to iterate different designs," with P9 elaborating that "with the plugin it was a lot easier. It helped with making things a lot quicker. Without the plugin, I was kind of stressed because I put high expectations on myself and then I do not see the result."

The remaining three participants were ambivalent as to whether the plugin made exploration easier: P12 stated that the conditions' workflow offered the same level of ease but in different ways, and P4 and P15 specified that the extra effort of seeking satisfactory designs within the output made up for the greater ease over creating these designs by hand.

Numerous designers commented on how the tool helped them explore via **designs' visualization** at the click of a button. P16 said, "I have to see it as a full product; of course, the colorization was helping me in visualizing different options" and noted, "Even before thinking of what I want to display, I get some prompts. And I can build up my ideas on what I have already seen works visually, or it conveys the message I want to convey." Visualizing given colors in the design was cited as another useful feature: "I usually have ready-made palettes [...] but if I had to design a new screen or poster, the plugin might be good to test this with those"(P11).

Many participants treated the plugin as a source of **inspiration** (e.g., P7 said, "It can show you different options. You can glance at them and get some ideas and know where you can take it to"). Some designers even exploited it to express their bolder side on the canvas: P12 was "inspired to try out bolder options," and P14 felt able to be "more radical and bolder with color choice."

Another factor cited as facilitating exploration was the **quantity of options**. P2 stated that "it gives a lot of examples; it gives a lot of exploration," and P9 thought that CoColor's "feature of seeing many, different options would be really helpful when I do designs." A few designers stated that, with such a vast quantity of outputs, some designs' were subpar. Several participants found this useful, though, in that seeing ideas to rule out altogether saved time – "even if something is not looking good, you see it, and that visually gives you a cue that this thing might not look good instead of trying it out first" (P2).

Participants mentioned getting fixated on some approaches in the plugin's absence, from having already envisioned the design in one way before beginning. In such cases, P12 found, "early exploration with the plugin avoids getting fixated." It gave **support for flexibility**.

6.2.2 Workflow. One aspect of the **difference in workflow** was the sequence of edits for exploring colors. Whereas P12 described having to "try a certain color on a certain part of the design, then there is a snowball effect of me trying it also on another one and then another one, and then I got ideas" in the no-plugin setting, "when using the plugin, I was already presented the options of where each color would be, and then all I needed to do was little tweaks here and there on the options presented." Summarizing another difference noted by several designers, P3 said that work "without the plugin requires more of thinking about what I would try to do in advance." That is, one had to consider what one wanted to do first. Several designers found editing the colors by hand faster – if they already had a vision. Elaborating on this, P6 described preferring a combined approach: coloring the parts for which she had a vision and letting the plugin color the rest.

This approach reflected a few designers' sense of the plugin's solid **integration with Figma** features. For instance, P11 found that "you can choose [options], and then you can refine them, using the Figma design functionalities. I think that is the best way, to combine those."

Some participants praised the plugin for **enjoyability of use**. P14 called its process "much more fun" for freeing the designer's time for exploring and visualizing as opposed to changing the colors of all elements merely for validation of some ideas. Echoing this sentiment, P16 said, "It was fun to explore, and it was enhancing my creativity, while without the plugin I was just doing my homework." Throughout the no-plugin condition, in contrast, P14 was antsy: "I felt all the time 'I really want to use it.' I wanted to try it out and see what it shows me."

6.2.3 Functionality. Most features were well-liked, with some designers going as far as to say that "all of the features make sense" (P14) and would mesh well with their practice (P16). Participants especially appreciated being in control at key points in the pipeline, where the plugin let them make decisions and edit the outputs before the next step.

The majority valued the **colors' extraction from focal objects**, and many regarded this function as aligned well with their typical workflow: "That is what I would do manually as well, collect some inspirational images that I like and pick colors from these" (P3). Regarding the latter task, P10 even claimed that "when i was trying to pull out colors from mood images, this would be the perfect tool." When asked which features fit their work style best, participants cited **coloring based on palettes** the most frequently. They found it "very useful" (P1) because "it saves a lot of work" (P3), and P6 said it "would also fit well if I have an idea of a palette."

Also, the **palettes' quality** was appreciated by P16, who found most of them "pretty balanced" and went on to state that "you had

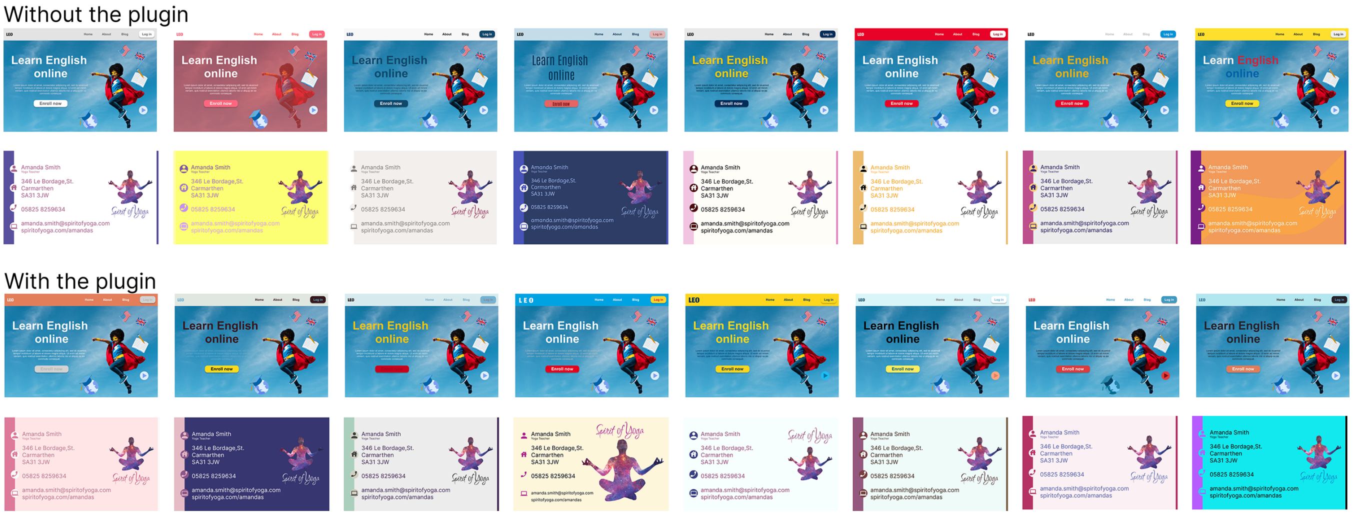

Figure 17: Design outcomes from the user study's task set 1, without and with the plugin. Of the several designs created, this figure shows the one given the highest rating by its designer.

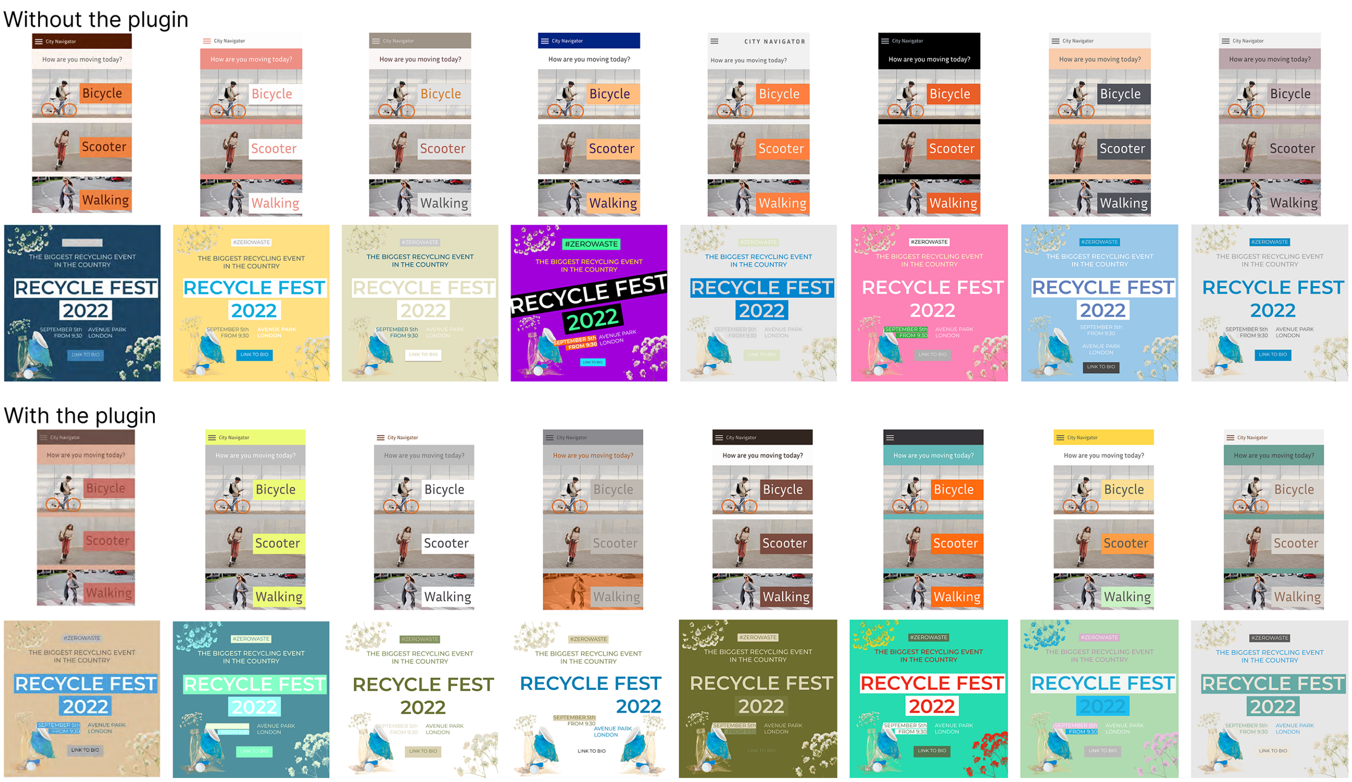

Figure 18: Design outcomes from task set 2.

a contrast color, a more light color and accent colors." On the other hand, P14 had expected color-wheel-based templates so perceived the palettes in the plugin as somehow "off." In a similar vein, P2 felt that the palettes were "not based on art styles." While the colorizations' value for visualization and exploration was appreciated, P3 and P15 criticized the **colorization quality** with respect to accessibility (P3: "I felt the colors that I got weren't really accessible [...]. That created the need to adjust the designs quite a lot").

Some participants wanted greater **diversity of suggestions**. In their view, similarity of some palettes led to similarity of designs. In particular, P10 and P8 seemed to find several palettes too similar and commented that this resulted in less varied design suggestions. P3 drew attention to the issue when realizing that the favorite designs chosen "were all similar."

Although the **extent of features** satisfied most subjects, some expressed a desire for still more control over the palettes. P14 requested "saving palettes," for the plugin's integration into long-term projects – alongside a function for their deletion. After a decision not to use it, a palette becomes "clutter" (P14) or "noise" (P1), increasing the browsing and computation time needed.

6.2.4 Speed and Transparency. Six participants found the plugin overly slow and that the **computation time** hampered their progress. P14 stressed, "The slowness was for me the biggest turnoff. Otherwise [..] I would probably always use it in the beginning in my designs."

Finally, some musings addressed whether the algorithms behind the plugin should have more **transparency** – "somehow it would be nice to see more [of the] logic behind it; now, everything is happening in the background" (P14). In fact, for P3, the opaque nature of color assignment made it "hard to create my own palettes because it was hard for me to see what is what." P7 too noted the assignment's predictability but ultimately concluded that "maybe it is, at the end of the day, irrelevant. The cognitive load of using it would be more."

6.3 Coloring Outcomes

Visual inspection of the outcomes shows no obvious difference in the color-design options' quality, an assessment that aligns with designers' own evaluation of the outcomes as discussed above. Figure 17, presenting each designer's highest-rated outcome from each case in set 1, and Figure 18, for set 2, show that participants generated a rich range of ideas in both conditions. Some colors are identical across conditions, mainly those contained in the images (designers frequently picked colors from the images via the pipette tool); however, the palette-extension functionality clearly brought in colors that designers exposed to the no-plugin condition did not try. Examples are the yellow in the navigation app and the mint green for the yoga business card in.

7 SUMMARY AND FUTURE WORK

The novel interaction technique presented in this paper successfully augments human creativity to produce well-designed coloring schemes for graphic designs. As intended, the proposed technique improved the selection of the color schema and the generation of fully colored designs:

(1) Designers found that exploring color themes became easier. Automated transiting between the three color design spaces and being able to visualize various options without as much manual editing helped them evaluate the range of options and seek inspiration for what to try next.
(2) The study's participants regarded the three-step process supported by our interaction technique as intuitive. Practitioners valued the features supported on every level and made use of them.

We received several constructive suggestions for improvement pertaining to details of the implementation and possible additional features, such as stronger filtering (for genuinely unique results), steering of the color-assignment possibilities, greater transparency, saving and deletion of palettes, increased breadth of palette options, more power for selection of the focal object (e.g., importing "mood images" or generating palettes by means of words), better alignment with designer-specific preferences, less clutter, and evaluation of colorization specialized for GUIs and graphical design.

We did not attempt to measure computational performance. Our study was assuming a short working effort for designers to explore and refine the results. Participants cited slow computation performance as the biggest drawback of the current implementation. The worst bottleneck arose from slow computations in the palette-extension step. Considerably slow evaluation via the GMM led to long computation times since the technique's MCTS uses this model frequently. Recoloring a large quantity of colorizations is another lengthy process.

Irrespective of our coloring algorithms' contrast checks, not all suggestions demonstrated great enough contrast to make all information visible for a wide audience. Most designers in our study detected this issue and corrected the colors for guaranteed legibility as was necessary. Although our study allowed us to evaluate CoColor for exploration of color schemes, removing low-contrast options from among the choices would increase efficiency by eliminating clutter and any need for manual corrections. Furthermore, there are ethics concerns in recommending inaccessible color designs to a wider group of users who might use them without questioning or editing them.

We envisage several algorithmic and computational techniques to address these concerns. 1) Assignment could be optimized to render only those options that are both distinct and accessible. This would not only save on computation time but also improve the interaction by removing clutter. 2) Prioritizing computation of those coloring schemes most likely to be desirable could be facilitated: the implementation could learn designer preferences or understand the semantics of colors and focal objects beyond saliency. 3) Finally, dedicated deep-reinforcement-learning models could function in place of the current set of generalized algorithms. For instance, a learned policy model could bring greater efficiency to exploring the palette space.

ACKNOWLEDGMENTS

This work was funded by Huawei Technologies. Antti Oulasvirta was funded by Finnish Center for Artificial Intelligence, an Academy of Finland flagship.

REFERENCES

[1] Dan S. Bloomberg and Leptonica. 2008. Color quantization using modified median cut. https://www.semanticscholar.org/paper/Color-quantization-using-modified-median-cut-Bloomberg-Leptonica/d246923c5d559445b4d699d6fe413895250156d3
[2] John Brooke. 1996. SUS: A "quick and dirty" usability scale. In *Usability Evaluation in Industry*. 189–194.
[3] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, and Gregg Vanderheiden. 2008. Web content accessibility guidelines (WCAG) 2.0, WWW Consortium (W3C).
[4] Erin A. Carroll, Celine Latulipe, Richard Fung, and Michael Terry. 2009. Creativity factor evaluation: Towards a standardized survey metric for creativity support.

In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*. ACM, 127–136.

[5] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. 2008. Monte-Carlo Tree Search: A new framework for game AI. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 4, 1 (2008), 216–217.

[6] Nigel Cross. 1982. Designerly ways of knowing. *Design Studies* 3, 4 (1982), 221–227.

[7] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual Symposium on User Interface Software and Technology (UIST '17)*.

[8] Zhenyu Gu and Jian Lou. 2016. Data driven webpage color design. *Computer-Aided Design* 77 (2016), 46–59.

[9] Shunan Guo, Zhuochen Jin, Fuling Sun, Jingwen Li, Zhaorui Li, Yang Shi, and Nan Cao. 2021. Vinci: An intelligent graphic design system for generating advertising posters. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Article 577.

[10] Sandra G. Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 50. SAGE, Los Angeles, CA, 904–908.

[11] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in Psychology*. Vol. 52. Elsevier, 139–183.

[12] Xiaodi Hou and Liqing Zhang. 2007. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2007.383267

[13] Guosheng Hu, Zhigeng Pan, Mingmin Zhang, De Chen, Wenzhen Yang, and Jian Chen. 2014. An interactive method for generating harmonious color schemes. *Color Research & Application* 39, 1 (2014), 70–78.

[14] Shanshan Huang, Xin Jin, Qian Jiang, and Li Liu. 2022. Deep learning for image colorization: Current and future prospects. *Engineering Applications of Artificial Intelligence* 114 (2022), Article 105006.

[15] A. Hurlbert and Y. Ling. 2012. Understanding colour perception and preference. In *Colour Design*. Elsevier, 129–157.

[16] Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, Nathan Ong, and Adriana Kovashka. 2017. Automatic understanding of image and video advertisements. In *2017 IEEE International Conference on Computer Vision (ICCV 2017)*. IEEE, 1705–1715.

[17] Johannes Itten. 1970. *The Elements of Color*. Vol. 4. John Wiley & Sons.

[18] Alexander Ivanov, David Ledo, Tovi Grossman, George Fitzmaurice, and Fraser Anderson. 2022. MoodCubes: Immersive spaces for collecting, discovering and envisioning inspiration materials. In *DIS '22: Designing Interactive Systems Conference*. ACM, 189–203.

[19] Ali Jahanian, Jerry Liu, Qian Lin, Daniel R. Tretter, Eamonn O'Brien-Strain, Seungyon Lee, Nic Lyons, and Jan P. Allebach. 2013. Automatic design of colors for magazine covers. In *Imaging and Printing in a Web 2.0 World IV*, Vol. 8664. International Society for Optics and Photonics, SPIE, 55–63.

[20] Ghita Jalal, Nolwenn Maudet, and Wendy E. Mackay. 2015. Color portraits: From color picking to interacting with color. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 4207–4216.

[21] EunJin Kim and Hyeon-Jeong Suk. 2016. Key color generation for affective multimedia production: An initial method and its application. In *MM '16: Proceedings of the 24th ACM International Conference on Multimedia*. ACM, 1316–1325.

[22] EunJin Kim and Hyeon-Jeong Suk. 2017. Thoughts and tools for crafting colors: Implications from designers' behavior. In *DIS '17: Proceedings of the 2017 Conference on Designing Interactive Systems*. ACM, 321–331.

[23] Hye-Rin Kim, Min-Joon Yoo, Henry Kang, and In-Kwon Lee. 2014. Perceptually-based color assignment. *Computer Graphics Forum* 33, 7 (2014), 309–318.

[24] Janin Koch, Nicolas Taffin, Michel Beaudouin-Lafon, Markku Laine, Andrés Lucero, and Wendy E. Mackay. 2020. ImageSense: An intelligent collaborative ideation tool to support diverse human–computer partnerships. *Proceedings of the ACM on Human–Computer Interaction* 4, CSCW1 (2020), Article 45.

[25] J. Kopacz. 2012. Enhancing design using colour. In *Colour Design*. Elsevier, 243–270.

[26] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. 2014. Deep Gaze I: Boosting saliency prediction with feature maps trained on ImageNet. https://doi.org/10.48550/ARXIV.1411.1045

[27] Sharon Lin and Pat Hanrahan. 2013. Modeling how people extract color themes from images. In *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3101–3110.

[28] Sharon Lin, Daniel Ritchie, Matthew Fisher, and Pat Hanrahan. 2013. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Transactions on Graphics* 32, 4 (2013), Article 37.

[29] Akis Linardos, Matthias Kümmerer, Ori Press, and Matthias Bethge. 2021. DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 12919–12928.

[30] Barbara J. Meier, Anne Morgan Spalter, and David B. Karelitz. 2004. Interactive color palette tools. *IEEE Computer Graphics and Applications* 24, 3 (2004), 64–72.

[31] Nicolas Mellado, David Vanderhaeghe, Charlotte Hoarau, Sidonie Christophe, Mathieu Brédif, and Loic Barthe. 2017. Constrained palette-space exploration. *ACM Transactions on Graphics* 36, 4 (2017), Article 60.

[32] B. Mikellides. 2012. Colour psychology: The emotional effects of colour perception. In *Colour Design*. Elsevier, 105–128.

[33] Sebastian Montabone and Alvaro Soto. 2010. Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image and Vision Computing* 28, 3 (2010), 391–402. https://doi.org/10.1016/j.imavis.2009.06.006

[34] A. H. Munsell. 1912. A pigment color system and notation. *The American Journal of Psychology* 23, 2 (1912), 236–244. http://www.jstor.org/stable/1412843

[35] Naila Murray, Luca Marchesotti, and Florent Perronnin. 2012. AVA: A large-scale database for aesthetic visual analysis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. ACM, 2408–2415. https://doi.org/10.1109/CVPR.2012.6247954

[36] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2011. Color compatibility from large datasets. *ACM Transactions on Graphics* 30, 4 (2011), Article 63.

[37] Li-Chen Ou and M. Ronnier Luo. 2006. A colour harmony model for two-colour combinations. *Color Research & Application* 31, 3 (2006), 191–204.

[38] Li-Chen Ou, Ming Ronnier Luo, and Guihua Cui. 2008. A colour design tool based on empirical studies. In *Undisciplined! – DRS International Conference 2008*. Elsevier, Article 66.

[39] Huy Q. Phan, Hongbo Fu, and Antoni B. Chan. 2017. Color orchestra: Ordering color palettes for interpolation and prediction. *IEEE Transactions on Visualization and Computer Graphics* 24, 6 (2017), 1942–1955.

[40] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane, and Martin Jagersand. 2020. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition* 106 (2020), Article 107404. https://doi.org/10.1016/j.patcog.2020.107404

[41] Alexander Sage, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. 2017. LLD – Large Logo Dataset – version 0.1. https://data.vision.ee.ethz.ch/cvl/lld.

[42] Maria Shugrina, Amlan Kar, Sanja Fidler, and Karan Singh. 2020. Nonlinear color triads for approximation, learning and direct manipulation of color distributions. *ACM Transactions on Graphics* 39, 4 (2020), Article 97.

[43] Maria Shugrina, Jingwan Lu, and Stephen Diverdi. 2017. Playful palette: An interactive parametric color mixer for artists. *ACM Transactions on Graphics* 36, 4 (2017), Article 61.

[44] Maria Shugrina, Wenjia Zhang, Fanny Chevalier, Sanja Fidler, and Karan Singh. 2019. Color Builder: A direct manipulation interface for versatile color theme authoring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Article 456.

[45] Hossein Talebi and Peyman Milanfar. 2018. NIMA: Neural Image Assessment. *IEEE Transactions on Image Processing* 27 (2018), 3998–4011.

[46] Garreth W. Tigwell, David R. Flatla, and Neil D. Archibald. 2017. ACE: A colour palette design tool for balancing aesthetics and accessibility. *ACM Transactions on Accessible Computing* 9, 2 (2017), Article 5.

[47] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and explore with a layout optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, 543–555.

[48] Masataka Tokumaru, Noriaki Muranaka, and Shigeru Imanishi. 2002. Color design support system considering color harmony. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*, Vol. 1. IEEE, 378–383.

[49] Anne Tomes, Caroline Oates, and Peter Armstrong. 1998. Talking design: Negotiating the verbal–visual translation. *Design Studies* 19, 2 (1998), 127–142.

[50] Arthur Robert Weeks and G. Eric Hague. 1997. Color segmentation in the HSI color space using the K-means algorithm. In *Nonlinear Image Processing VIII*, Vol. 3026. SPIE, 143–154.

A THE QUESTIONNAIRE ANSWERS

Figure 19 to 23 visualize the responses of the study participants to the questionnaires included in the user study.

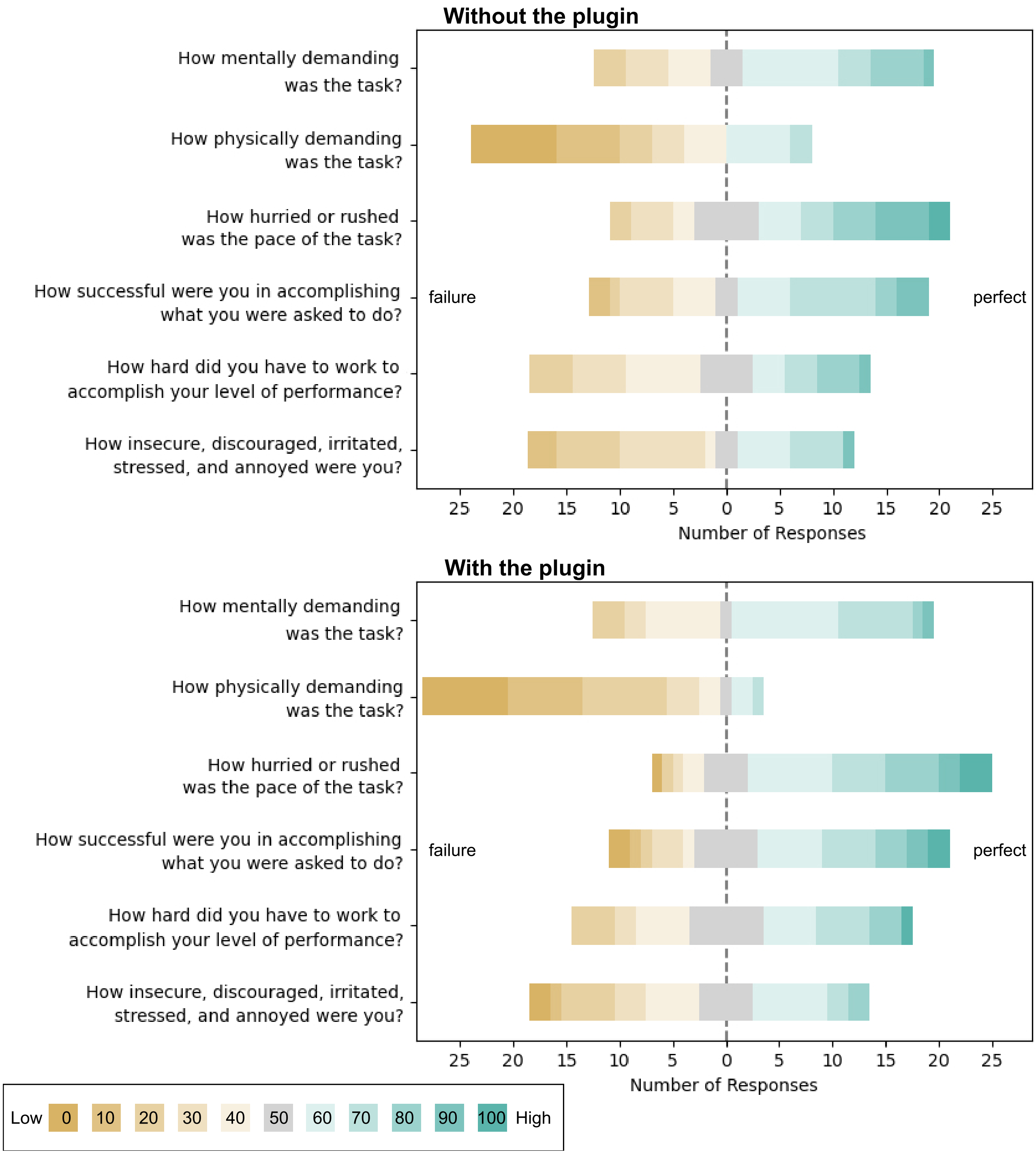

Figure 19: Responses to the NASA TLX questions after each task.

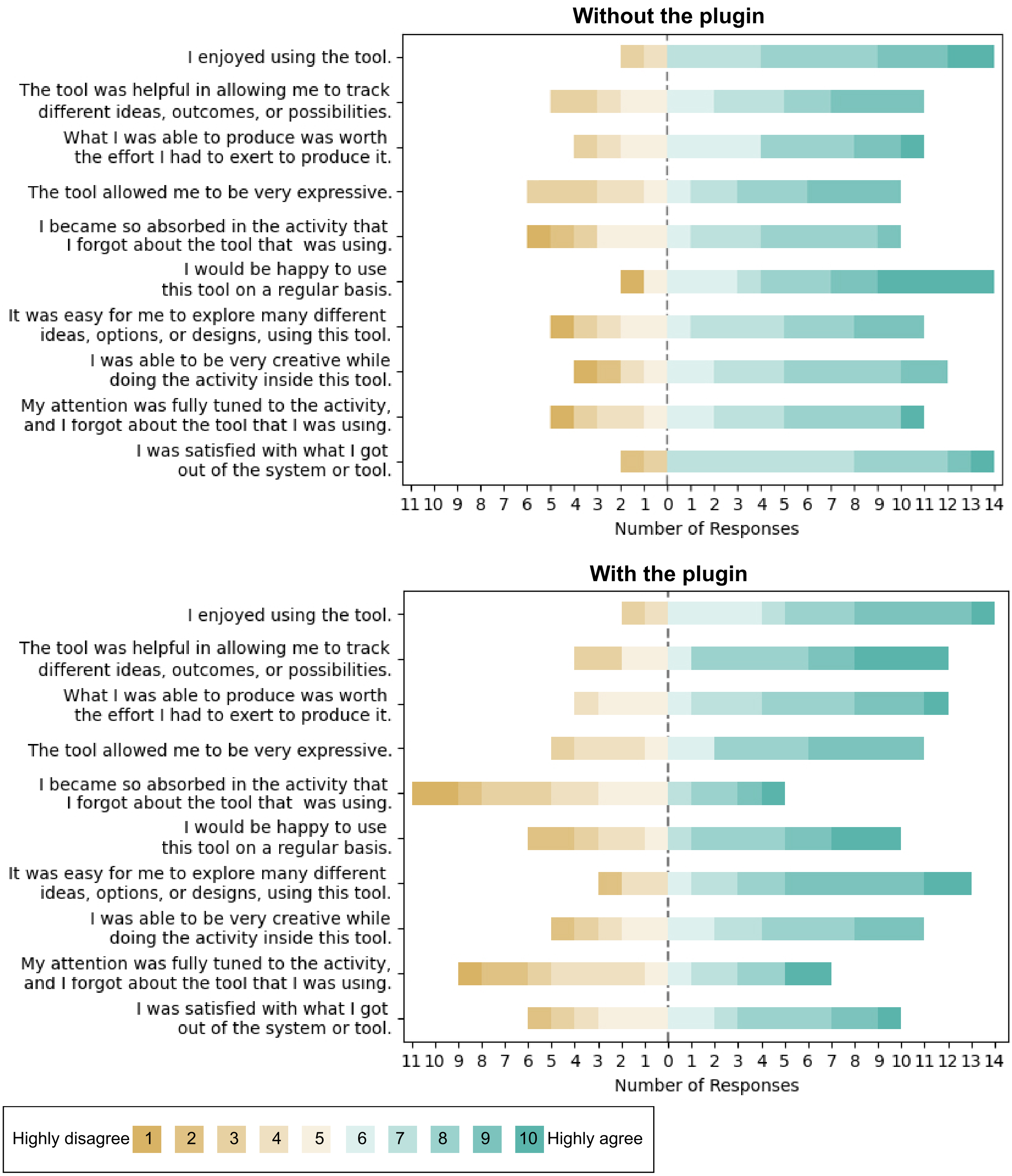

Figure 20: Responses to the CSI questions after exposure to the no-plugin and to the plugin condition.

Figure 21: Responses to the SUS questions after the no-plugin and the plugin condition.

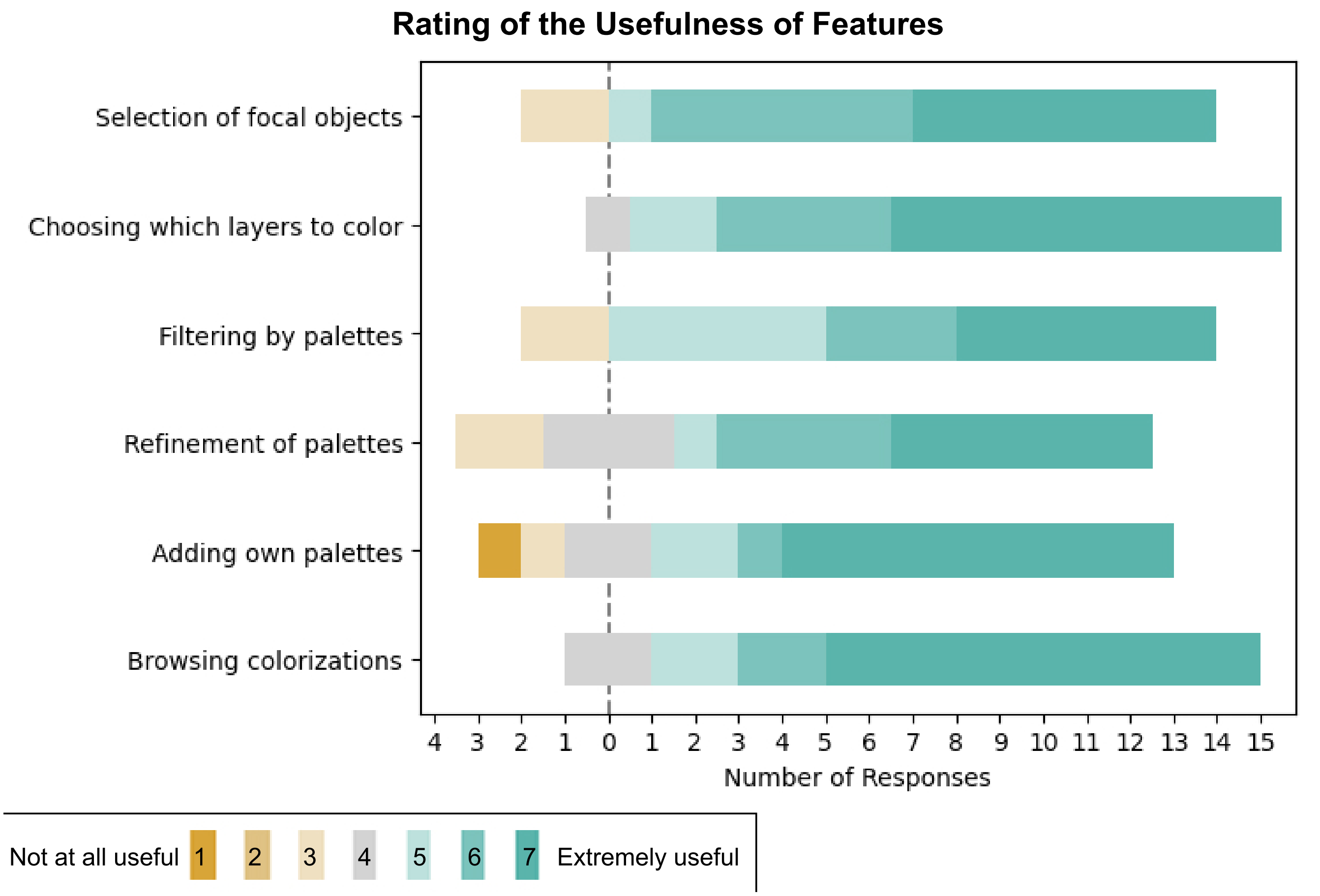

Figure 22: Ratings for the usefulness of the plugin's various features.

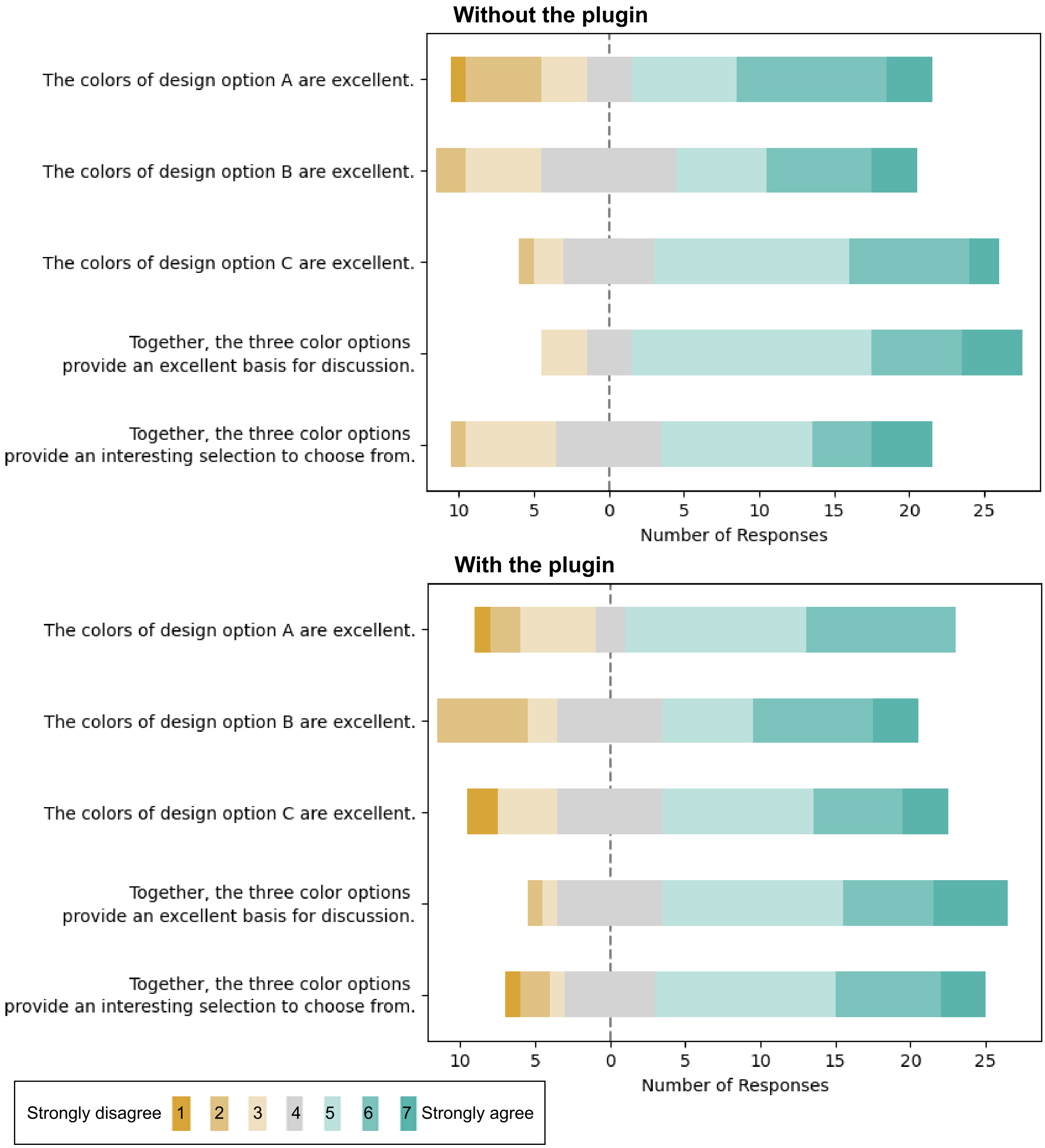

Figure 23: Participants' evaluation of the three design options (A, B, and C) that they created during each task.