Real-Time Color Palette Generation: Enhancing Design Efficiency

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Jenny Schilling

Spring, 2025

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Briana Morrison, Department of Computer Science

Real-Time Color Palette Generation: Enhancing Design Efficiency

CS4991 Capstone Report, 2024

Jenny Schilling
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
xdj3kg@virginia.edu

ABSTRACT
In the digital design industry, the challenge of quickly generating and applying aesthetically pleasing color palettes to digital products remains a significant hurdle for designers. To address this, I developed a real-time color palette generation tool during my internship with a popular sportswear company, leveraging both color science and software engineering principles. This project utilized a well-known painting and texture software, while integrating machine learning algorithms with Python such as *k*-means clustering to extract dominant colors from reference images and apply them to design components. The resulting tool not only improved efficiency in the design process, but also enhanced creativity by offering designers more experimental control over color application. While the initial implementation was highly successful, future work is needed to refine the color extraction algorithms, expand the tool's customization options, and conduct comprehensive testing for broader industry applications.

1. INTRODUCTION

How can aspects of computer science and machine learning be utilized to enhance the digital design workflow? A designer's job is to create unique, visually appealing products for the consumer. These products are often considered works of art in the eye of the designer, with many considering the design process an intimate experience that tells a story. Additionally, many designers use reference images to shape this storytelling, allowing for a more seamless starting point. Optimizing the efficiency of this design process would be highly beneficial to the workflow of product development. This was the primary motivation for my project: identifying inefficiencies within the company's design workflow and using machine learning to improve it, giving designers more time to focus on their creative work.

A crucial aspect of the design process, a color palette is a collection of colors that work together to give products a cohesive and uniform aesthetic. Traditionally, selecting these palettes has been a manual and time-consuming process, heavily reliant on the designer's experience. However, this subjective nature can lead to unwanted variability and inconsistencies, especially when working with large-scale applications.

To tackle this problem, I took advantage of machine learning to automate the color selection process by employing *k*-means clustering. Dominant colors were extracted from reference images and applied to design palettes and components. This automation allows designers to establish a theme while ensuring consistency across various visual elements. By blending automation with customization, the tool enhances artistic control, improves the efficiency of the design process, and supports scalability, which is crucial as the digital design industry continues to grow with consumer demand.

2. RELATED WORKS

One of the most closely related pieces of work to my project is the autonomous color extraction method using saliency algorithms developed by students at Purdue University [2]. This research aimed to improve traditional color extraction methods by focusing not just on dominant colors but also on those that are the most "eye-catching" to human perception. They employed a technique using color saliency, which involves converting the sRGB (standard red, green, and blue color space) values from an image to the CIELAB color space. CIELAB (or L*ab**) defines color based on three values: lightness, chroma, and hue, making it more suited for numerical comparison of color similarity from a human visual perspective.

Their approach considers both aesthetic appeal and the constraints of color harmony. Using a method based on *k*-means clustering, the algorithm computes a saliency map, extracts the *k* most salient pixels, and uses this set of dominant colors that balances visual contrast with the need for a cohesive design. By introducing a saliency factor into their clustering algorithm, Jahanian, et al. provide a solution to ensure that the resulting palettes are diverse, visually engaging, and align with how humans perceive the color in an image.

Another notable contribution in the field of color palette generation is the work done by Lin and Hanrahan [5], who developed a method for extracting color themes from images using a regression model trained on themes created by real people. Their system was created to assist designers by providing an interface that suggested harmonious colors based on user input. They applied principles from color theory, such as complementary and analogous color schemes, but also incorporated perceptual uniformity by using the CIECAM02 color appearance model. The system optimized the color palette by minimizing perceptual error, ensuring that the selected colors appeared balanced and visually pleasing to users. This method allows designers to interact with the palette generation process, ensuring that the final product meets both the functional requirements and the aesthetic goals of the project.

Lin and Hanrahan's work introduced a very interesting element of this user-driven control, which contrasts with fully automated systems but highlights the importance of designer input in the creative process. Their approach influenced my project in the sense that they demonstrated how algorithmic color selection can be facilitated by interactive elements.

These larger works highlight how machine learning and algorithmic techniques can be used to automate traditionally subjective tasks like color selection, which has influenced the development of my own approach. By leveraging a mix of clustering algorithms, saliency maps, and user input, this project aims to further streamline the digital design workflow, providing a tool that balances artistic control with computational efficiency.

3. PROJECT DESIGN

The purpose of this project was to create a single tool that addresses color palette creation inefficiencies by automatically extracting the dominant colors from a reference image using machine learning and color science techniques. The process towards completion can best be outlined in three key components: the extraction algorithm, post-processing, and integration with the design software.

3.1 Color Extraction Algorithm

Arguably the most difficult part of this project, the algorithm to extract the dominant colors from an image was the core of the design process. In short, it uses *k*-means clustering to identify a user-specified number of dominant colors from the RBG pixel values of an image. Though other color extraction techniques, such as color quantization, were explored, *k*-means was chosen for its simplicity and, most importantly, efficiency.

The first step of the algorithm is the preprocessing phase, in which the image is resized and converted to a uniform color space (sRGB, in this case) [4]. Essentially, every pixel gets mapped to a tuple containing its red, green, and blue color values (from 0-255). Additionally, when the image is resized, I opted to reduce the image quality to that of 15% its original number of pixels to ensure the algorithm has less data to sort through, but the color integrity is preserved.
The image data is subsequently fed into the *k*-means clustering algorithm, where the user selects the number of clusters (k) to define how many dominant colors to extract. The algorithm then randomly selects k initial centroids in the color space, and each remaining pixel in the image gets assigned to its nearest centroid. The distance between a pixel and a centroid is calculated using Euclidean distance in the sRGB color space, so pixels that are close together in color will be assigned to the same centroid, forming a cluster of similar colors.

Next, the algorithm recalculates the position of each centroid by averaging the colors of all the pixels in its respective cluster, moving the centroids toward the "center" of their clusters, representing the average color of that group of pixels. This process of assigning pixels to clusters and updating centroids is repeated iteratively until the centroids no longer shift significantly, and thus "convergence" is reached. Once the algorithm converges, the final centroids represent the dominant colors extracted from the image.

3.2 Post-Processing

Through testing, I determined that the k extracted colors from the image were not reliably what we, as humans, would identify as the most dominant colors. Take the example of an image of a large red wall with a small blue door. If the user specifies they want to extract two colors, the algorithm will return a palette of two different types of reds, as opposed to two *different* colors, like a red and a blue. I found the best method to address this was to always extract the 20 most dominant colors from the image (i.e. set k=20), and then use an alternative process to reduce the palette down to the user-specified number of colors. 20 was found through experimentation to be a number that would consistently extract diverse colors from images, but in a timely manner. The reduction procedure involved converting the 20 extracted colors from the sRGB color space to the CIELAB color space, which is more useful in comparing colors on a scale of human-perception. Figure 1 shows this color space in 3D, where the L* represents lightness or darkness, a* stands for redness or greenness, and b* indicates the yellowness or blueness of a color.

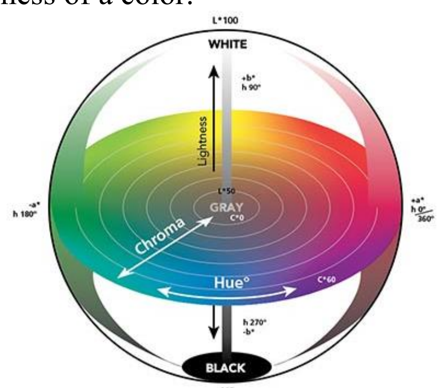

Figure 1: Colorimetric Coordinates in CIELAB Models [6]

Using this model, we can use what is essentially a modified distance formula to calculate the "distance" or visual difference between two colors. This value is termed as delta E, with the formula defined by:

$$\Delta \boldsymbol{E} = \sqrt{(\Delta \boldsymbol{L}^*)^2 + (\Delta \boldsymbol{a}^*)^2 + (\Delta \boldsymbol{b}^*)^2}.$$

When the palette of 20 colors is originally generated, they are inherently sorted by most-to-least dominant in the image. Using this order, the palette is reduced by computing the ΔE between each color, finding the two colors with the smallest ΔE, and removing the less prevalent of those two colors. This is repeated until only the user-specified number remain, ensuring that the colors that are generated are the most dominant, but also the most unique in the image.

3.3 Integration with Design Software

The extracted color palette was then integrated into a widely used design software through a Python-based plugin, allowing designers to import reference images and generate color palettes directly within their workflow. This seamless integration enables designers to match extracted palettes to existing colors in the company's schema without leaving their familiar working environment.

The plugin interface includes options to customize the process, such as selecting the number of colors and choosing seasonal palettes. Once generated, designers can apply the extracted colors to models either by randomly assigning colors or preserving original color blocking patterns, supporting efficient and flexible design customization.

4. RESULTS

The development of the color palette generation tool resulted in a noticeable increase in efficiency for designers by taking the manual labor out of color selection and matching to the company's existing palette. Rather than designers having to spend time sifting through reference images and eye-balling consistency across colors, designers could now automatically work through the color selection stage. A quantitative analysis on the time to create palettes was not performed, but through qualitative feedback, there is significant evidence of the tool freeing up more room for creative thinking. Furthermore, this step not only sped up project timelines, but also cut down on potential mismatches that might sneak in when picking colors by eye.

Beyond just saving time, the tool was able to boost designers' creative processes. The randomized color applications and auto-generated palettes pushed designers to work outside the box and try combinations they likely would not have thought of manually. This ability to randomize color assignments or stick to the original product's color blocking gave designers the flexibility to enhance or maintain the appearance of their projects, making the design process feel more exploratory and less restrictive.

Finally, it streamlined workflow for populating and clearing the workbench palette, making it easy to apply changes and undo them. The integration with the design software allowed the workflow to operate more smoothly, letting designers transition between different stages without faulter. All these changes added up to a much more fluid process, whether that be from the initial brainstorming to the final tweaks, ultimately raising the bar for both speed and quality.

5. CONCLUSION

The importance of this project lies in increasing the efficiency of digital designers' workflow—streamlining the process of creating coherent, meaningful color palettes that can be applied seamlessly to designs or extracted from a brand's existing theme. To do this, features were added to a common 3D modeling software that allow designers to right-click a reference image, select parameters for extraction, and instantly obtain the dominant colors from the image using a modified *k*-means clustering algorithm. Furthermore, this functionality ensures aesthetic consistency across designs, reducing manual effort and encouraging creativity. Beyond the technical outcomes, this project also deepened my personal understanding of color science and its applications in software engineering, providing valuable insights into design automation. Ultimately, the primary takeaway is a tool that not only saves time but fosters creative freedom for the digital design industry.

6. FUTURE WORK

When considering building on the success of this project, several avenues for future work could further the tool's functionality and scope of implementation. First, integrating an adaptive color adjustment feature, like that of the popular color palette generator designed by Fabrizio Bianchi at *Coolors*, could allow for additional palette customization and tweaking after it has been generated [1]. This would add flexibility to fine-tune colors based on specific guidelines, and potentially enable broader industry adoption.

Another area for expansion involves applying the color extraction algorithm to a graphical image region extraction process, enabling the tool to distinguish key visual elements within images. Inspired by the work done by Jardim, et. al., this would extend the project's usefulness in fields that require a more concentrated visual analysis, like digital marketing or product design, where it is important to isolate primary visual cues from extraneous information such as text [3].

REFERENCES

[1] Bianchi, F. 2013. Coolors. Retrieved November 12, 2024, from https://coolors.co

[2] Jahanian, A., Vishwanathan, S. V. N., and Allebach, J. P. 2015. Autonomous Color Theme Extraction from Images Using Saliency. In *Proceedings of SPIE, 9408, Imaging and Multimedia Analytics in a Web and Mobile World 2015*, 940807. https://doi.org/10.1117/12.2084051

[3] Jardim S, António J, Mora C. Graphical Image Region Extraction with K-Means Clustering and Watershed. *Journal of Imaging*. 2022; 8(6):163. https://doi.org/10.3390/jimaging8060163

[4] Lewis, X. 2021. Extracting colors from an image using K-means clustering. Medium. Retrieved June 20, 2024, from https://towardsdatascience.com/extracting-colours-from-an-image-using-k-means-clustering-9616348712be

[5] Lin, S., and Hanrahan, P. 2013. Modeling how people extract color themes from images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3101–3110. http://dx.doi.org/10.1145/2470654.2466424

[6] Mouw, T. 2018. Tolerancing Part 3: Color space vs. color tolerance. X-Rite Color Blog. Retrieved June 2024, from https://www.xrite.com/blog/tolerancing-part-3