S.T. Yau High School Science Award

Research Report

The Team

Name of team member: Jeslyn Chen
School: Palo Alto High School
City, Country: Palo Alto, USA

Name of supervising teacher: Zhengfei Kuang
Job Title: Ph.D. candidate
School/Institution: Stanford University
City, Country: Palo Alto, USA

Title of Research Report

Shape Control and Color Harmonization for Improving Logo Design with Stable Diffusion

Date

August 19, 2023

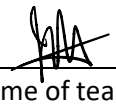
Commitments on Academic Honesty and Integrity

We hereby declare that we

1. are fully committed to the principle of honesty, integrity and fair play throughout the competition.
2. actually perform the research work ourselves and thus truly understand the content of the work.
3. observe the common standard of academic integrity adopted by most journals and degree theses.
4. have declared all the assistance and contribution we have received from any personnel, agency, institution, etc. for the research work.
5. undertake to avoid getting in touch with assessment panel members in a way that may lead to direct or indirect conflict of interest.
6. undertake to avoid any interaction with assessment panel members that would undermine the neutrality of the panel member and fairness of the assessment process.
7. observe the safety regulations of the laboratory(ies) where we conduct the experiment(s), if applicable.
8. observe all rules and regulations of the competition.
9. agree that the decision of YHSA is final in all matters related to the competition.

We understand and agree that failure to honour the above commitments may lead to disqualification from the competition and/or removal of reward, if applicable; that any unethical deeds, if found, will be disclosed to the school principal of team member(s) and relevant parties if deemed necessary; and that the decision of YHSA is final and no appeal will be accepted.

(Signatures of full team below)

X 

Name of team member:

X Zheng Fei Kuang

Name of supervising teacher:

Shape Control and Color Harmonization for Improving Logo Design with Stable Diffusion

Jeslyn Chen[1] and Zhengfei Kuang[2]

[1] *Palo Alto High School, CA, USA*
[2] *Computer Science Dept., Stanford University, CA, USA*

Abstract: AI-powered image generation models have become increasingly popular recently. They are able to create unique and aesthetically stunning visuals based on a given text prompt. However, the resulting images are often unexpected and unpredictable, and the user has little control over various details in the generated output. Producing a desired image would therefore require careful prompt crafting and lots of trial and error. In addition, it is fairly time-consuming to visually assess a large number of generated images. In this project, we focus on improving the workflow efficiency in logo generation using Stable Diffusion. We analyze the performance of different metrics in assessing the quality of logo images and their correlation to subjective evaluation. A good metric, such as CLIP (Contrastive Language-Image Pre-training), helps to automatically filter out poor candidates, thus saving time in reviewing a large number of images. We then incorporate ControlNet into the process of creating logos from a desired stretch for better spatial consistency. Further improvement is accomplished by extracting and analyzing color schemes, followed by identifying color harmonies to help separate the logos that are aesthetically superior from those that are inferior. By leveraging the results of color clustering, we also develop a solution to enable manipulation of colors in a logo image.

1. Introduction

The visual design of logos is in general aimed at delivering content that is engaging and impactful so that it effectively communicates and captivates viewers. Oftentimes, simplicity is crucial in terms of the layout, shapes, and composition of colors. Admittedly, it is a highly subjective process when people look at visual designs. Fortunately, there are well established theories and guidelines that artists follow when they make aesthetic choices in the details. There also exist various combinations in logo design, each with its own pros and cons. While it is not a strict rule, one of the general and interesting recommendations is to keep a simple palette without an excessive number of colors. Such a color scheme may offer several benefits in terms of memorability, versatility, clarity and cost effectiveness.

Recently, generative AI tools, such as Midjourney [1], DALL-E [2] and Stable Diffusion [3], have been widely used in visual design. These tools and applications nowadays play a significant role by offering various benefits and capabilities to create new content based on existing artwork, style and themes. These tools produce fresh and inspiring elements that can be incorporated into the process of art design. The AI-powered tools are amazingly capable of creating complex and stunning visuals, which can be customized through the theme specified by text descriptions and other inputs.

Despite significant advancements and remarkable capabilities, generative AI tools can still fail or produce undesirable results in many scenarios. Reportedly, there are numerous examples of outputs that mismatch an intended vision, have misaligned aesthetic interpretations, or produce unrecognizable images, which can sometimes be ludicrous. Figure 1 illustrates a few instances of such unpleasant results. Depending on the AI models and training datasets, over-fitting during training may lead to the generation of similar and repetitive samples. On the other hand, under-fitting can result in vague and distorted images.

It is quite efficient, in terms of machine time, to create a bunch of images using the tools. However, it is very time-consuming to subjectively evaluate all the images and shortlist the most preferable outcomes for further assessment and refinement. In our experiments, we observed the existence of both

Fig. 1: Examples of mismatching or unrecognizable results from Stable Diffusion: (left) prompt - *hipster logo design for a coffee shop called "Virginia Roast"* and (right) prompt - *logo for a sports team with a cobra head mascot looking right, maroon and white, vector logo, professional graphic design, no text.*

impressive and poor designs generated from a text prompt. It has become apparent that there is a desire to automatically filter out undesirable poor results from the first iteration. The goal of this process is to qualify a smaller set of useful candidates without going through tedious subjective assessment.

Color is an important element of most forms of visual design, including logos. Color features are often used to look for style trends in visual representation. Analysis of color data in design either focuses on single colors or model-free histogram representations. A long tradition of research in color theory has shown that the relationships between colors have a significant impact on our perception [4, 5]. The choices of varying color palettes sometimes exhibit less than ideal color harmonies. Color harmonies [4–6] refer to the color matching of two or more colors. When organized in an orderly and coordinated manner, they can make people feel happy and satisfied. It has been widely used in graphic design. As it demonstrates strong correlation with emotion, it is further applied in computer vision e.g., affective computing and semantic analysis. There was research to combine classical color harmony theory with machine learning models [7, 8]. Stable Diffusion does not yet provide options to specify or constrain a particular color scheme. When a shape or layout from the generation becomes satisfying, it will be a lot more efficient if the color scheme can be automatically identified. It is also highly desirable that without re-generating from a same prompt, the color choices can be efficiently modified so that a target color harmony can be achieved.

In this paper, we present a novel method for AI-generated logos using Stable Diffusion and ControlNet [9, 10], with a focus on evaluating image quality using CLIP [11, 12] and optimizing color schemes. We start by introducing the critical modules of Stable Diffusion utilized in our method for logo generation. We also describe the integration of ControlNet [9, 10], which adds significant value to Stable Diffusion by offering enhanced regulation and conditioning, especially when AI generation input is accompanied by alternatives other than a textual prompt. To evaluate the quality of the generated logo images, different measurement options have been employed. However, traditional image quality metrics have been found to lack the accuracy and confidence in providing a good correlation with subjective evaluation. To address this issue, we propose a two-stage logo generation method by applying CLIP (Contrastive Language-Image Pre-training) [11, 12] to an image in comparison with the original text prompt, which as a reinforcement feedback provides a better indicative outcome. This therefore helps to filter out the undesirable images. Furthermore, we emphasize the significance of color schemes in images, and propose methods to extract the dominant colors from a generated logo image. This is followed

by the identification of chromatic attributes and the recognition of one or more color harmonies in the image. We further provide an approach to easily manipulating the color scheme, which leverages the color clustering results.

In summary, we propose a novel method for AI-generated logos using Stable Diffusion and ControlNet, focusing on evaluating image quality with CLIP and optimizing color schemes. The main contributions of this paper are:

- We propose combining Stable Diffusion with ControlNet to enhance the regulation and conditioning of logo generation, and incorporating the use of CLIP for image quality evaluation to provide reinforcement feedback and filter out undesirable results.
- We propose an approach to extract dominant colors from a generated logo image, identify chromatic attributes, and recognize color harmonies. We introduce a method to manipulate the color scheme of the logo, leveraging color clustering results, thereby highlighting the significance of color schemes in logo design.

We validate our approach through extensive experiments and demonstrate the effectiveness of our method in generating high-quality, aesthetically pleasing logos with optimized color schemes. Our results show that the proposed method successfully meets the design requirements and preferences of users, offering an innovative solution for automated logo generation. We believe that our approach holds potential for further development in the field of AI-assisted graphic design.

2. Related Work

2.1. Image Generation

Image generation has seen significant advances in recent years through the development of generative adversarial networks (GANs) and diffusion models.

Early work on GANs [13] demonstrated how a generator and discriminator neural network could be trained in an adversarial process to produce novel images that appear realistic. Since then, numerous improvements have emerged. Wasserstein GANs [14] improved training stability. Progressive growing of GANs [15] grew layers incrementally to boost image quality. GigaGAN [16] is one of the largest and most capable GAN models developed recently.

Recently, generative AI instruments like Midjourney [1], DALL-E [2], and Stable Diffusion [3] have gained popularity in the field of visual design. Diffusion models take a different approach, adding noise to data and learning to reverse the diffusion process to generate images. DDPM [17] demonstrated high-fidelity image generation through this technique. Follow-up work like DDIM [18] improved sampling speed. Most recently, diffusion models like DALL-E 2 [19] have shown the ability to generate realistic and controllable images at high resolutions.

2.2. Image Color Harmony

Extensive studies in color theory have demonstrated that the interplay of colors substantially influences our perception. O'Donovan et al. [4] investigate color compatibility theories using large online datasets, tests human color preferences, and develops quantitative models to assess the quality of color themes. They apply the learned model to tasks in color design for enhancing existing themes and extracting themes from images. Shevell [5] also provides an comprehensive overview of color science. The selection of different color palettes occasionally displays different color harmony and color harmonies refer to the pleasing arrangement or interaction of two or more colors. Color harmony [4–6] is a well researched topic in color theory. Cohen-Or et al. [6] introduces a method for enhancing the color harmony in photographs or general images by automatically finding the best harmonic scheme for the image colors and gracefully shifting hue values to fit the scheme while considering spatial coherence among neighboring pixels. There was also research to combine classical color harmony theory with machine learning models. Nishiyama et al. [7] propose 'bags-of-color-patterns', which evaluates color harmony

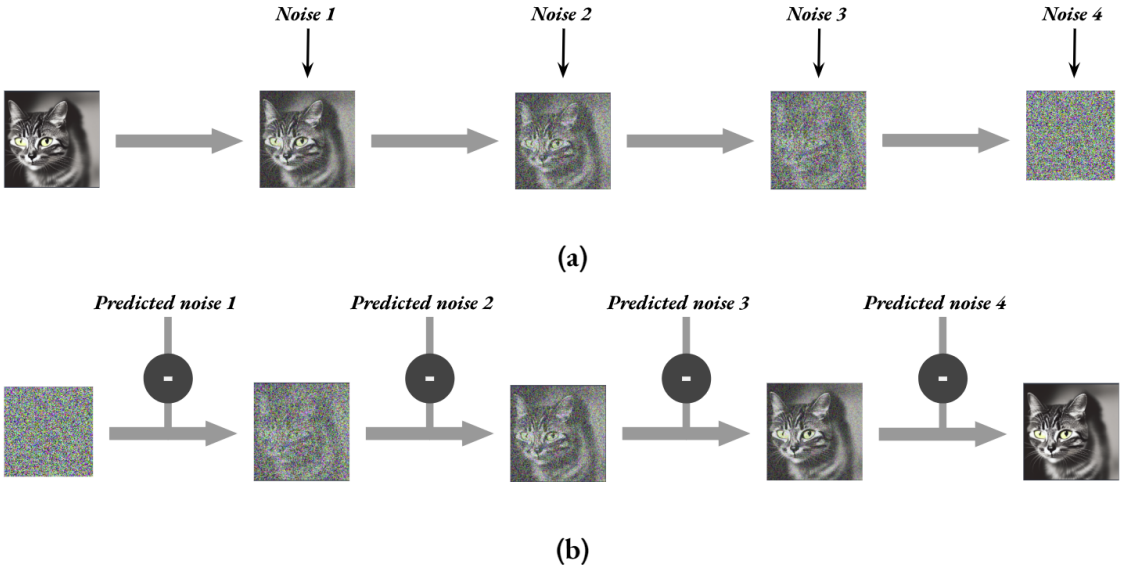


Fig. 2: (a) Noise is added at each step sequentially, and the noise predictor estimates the noise accumulated to each step. (b) Starting with a completely random image, it continuously subtracts the estimated noise from the original image.

in photos by analyzing local color patterns and improves aesthetic quality classification of photos. Lu et al. [8] propose a color harmony model for photo aesthetics assessment that uses Latent Dirichlet Allocation and considers spatial relationships between harmonious colors Tan et al. [20] introduce a highly efficient and scalable palette-based image decomposition algorithm that uses the geometry of images in RGBXY-space, allowing for real-time layer decomposition and interactive editing of the palette. Hue template-based models are often employed by both traditional and machine learning based color harmony methods for selecting color schemes to improve color [6–8]. Celebi et al. [21] investigates the performance of k-means as a color quantizer, and show that an efficient k-means implementation with the right initialization can be an effective color quantizer. Zheng et al. [22] introduce an adaptive K-means image segmentation technique, which involves utilizing the relationship between K values and connected domains for adaptive segmentation and achieves simplicity, accuracy, and effectiveness in its results. In this paper, we leverage the insights from existing color theory and color harmony research, and adopt the K-means clustering technique to effectively identify and manipulate dominant colors, to achieve harmonious color schemes in logo images.

3. Two-Stage Logo Generation with CLIP Quality Metrics

In this section, we will introduce the way to generate logos from conditional signals and filter high-quality generated samples automatically. We elaborate on the essential components of our Stable Diffusion model, namely the variational autoencoder (VAE), U-Net, and an optional text encoder. We discuss the process of conditioned image generation using text prompts and how it guides the noise predictor to create specific images. Additionally, we explain the application of the CLIP model to analyze the similarity between the generated image and the corresponding text prompt, leveraging its text and image encoder capabilities. Throughout the section, we provide illustrative figures to help visualize the described processes.

3.1. Diffusion-based Logo Generation

Stable Diffusion is a generative text-to-image AI tool that is one of the most advanced tools for visual content creation. One of the advantages of Stable Diffusion is that it is especially creative in producing new and different visual designs. However, it has issues with generating text and is difficult for the user

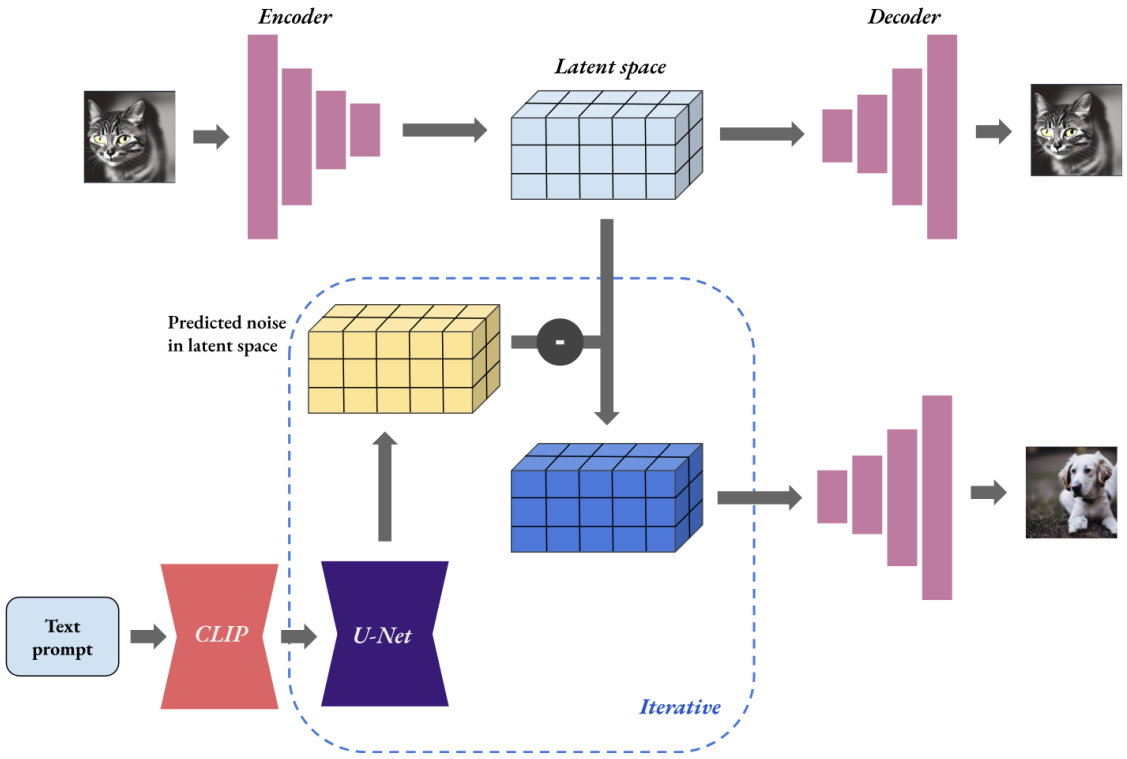


Fig. 3: Variational autoencoder (VAE) transforms the image to (by an encoder) and from (by a decoder) the latent space. The noise predictor U-Net takes in the latent noisy image and text prompt to estimate the noise in each iteration.

to control specific details of the resulting image, such as choices of colors. The latest announcement on SDXL 1.0 [23] described a lot of enhancements including more faithful text generation. It is able to create complex and aesthetically stunning images from only a few words. However, the creation of text remains frequently flawed in many testing reports. The generation of human hands is another significant challenge, which has been consistent across various AI-powered tools.

Stable Diffusion consists of 3 parts: the variational autoencoder (VAE), U-Net, and an optional text encoder [3, 11]. Here, we do not intend to overwhelm the readers with a lot of information. Rather, a high level description of individual modules is provided for completeness.

In Figure 2(a), it shows the process of forward diffusion. Noise is continuously added to a training image, and gradually turns the image into an uncharacteristic noise image. To reverse the diffusion, as illustrated in Figure 2(b), a neural network model is used to predict the noise added. The so-called noise predictor is the U-Net in Stable Diffusion architecture. So far the above description only covers the processes of unconditioned image generation. In other words, there is no control over the ultimate image, e.g., a cat or a dog. The purpose of conditioning is to steer the noise predictor so that the estimated noise at each step will produce something as intended, after its subtraction from the image.

In Stable Diffusion, the conditioning is addressed by the option of text prompt, which is processed and fed to the noise predictor. In Figure 3, the latent space is first illustrated. The forward and reverse diffusions are in fact done in the latent space, which represents an extremely compact form of image features. The VAE neural network includes both an encoder and a decoder. The CLIP model used to process the text prompt through modules of tokenizing, embedding, and transformer to generate the input for U-Net to steer image generation. The CLIP model itself is trained on large data sets of image-caption pairs and includes an image encoder and a text encoder. The image and the text embeddings are then trained to demonstrate their similarity in order to reach a point where at inference time the embedding of a text prompt is close to an image embedding that describes it.

The processes of predicting noise and subtracting the latent noise from the latent image are repeated for a number of sampling steps. When the latent image is finalized after all the interactions, the VAE decoder converts the latent image back to a representation in the pixel domain. In the example shown in Figure 3, it assumes a text prompt of generating a dog.

3.2. Evaluation on Image Quality Metrics

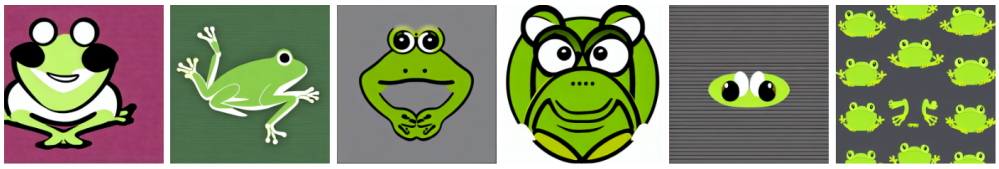
We first investigate different classical image quality metrics on generated logos. In this experiment, the following quality assessment methods are included: BRISQUE [24], CNNIQA [25], DBCNN [26], MUSIQ [27] and NIMA [28].

In the logo generation, there does not exist a reference for comparison purposes. Therefore, we select the above no-reference quality metrics. The five metrics are executed on all the images. In Tables 1-3, a selection of three prompts and their images are presented to demonstrate the visual appearance of the images in comparison with the reported quality metric scores.

Table 1: Test results of Prompt #1 – *lion emblem in a square, style of Clash of Clans, game icon*

	1	2	3	4	5
Prompt #1	Quality metrics				
Image	BRISQUE	CNNIQA	DBCNN	MUSIQ	NIMA
1	97.73	0.03	62.11	70.35	5.42
2	38.61	-0.17	56.83	67.46	5.06
3	29.38	0.28	67.38	75.42	5.49
4	13.27	-0.81	54.71	56.50	5.06
5	170.96	-0.19	62.33	69.31	4.69

Table 2: Test results of Prompt #2 – *vector graphic logo of frog, simple minimal, (-) realistic photo details*



	1	2	3	4	5	6
Prompt #2	Quality metrics					
Image	BRISQUE	CNNIQA	DBCNN	MUSIQ	NIMA	
1	198.04	-0.24	60.81	72.53	4.57	
2	130.39	1.35	75.46	75.35	5.58	
3	61.33	-0.24	60.41	73.13	4.83	
4	34.62	0.15	68.45	75.15	5.10	
5	130.07	0.20	69.10	73.58	5.63	
6	40.66	0.26	60.60	69.77	4.66	

We have observed that the results of image quality metrics are not always related to human visual assessment. Some logos with lower scores were more visually appealing than other logos with higher scores, and these inconsistencies existed across many different quality metrics. For example, the image numbers highlighted in red in Tables 1-3 are what we considered as visually unappealing. However, they would sometimes have higher scores reported by certain image metrics than other more aesthetic logos. There is a need to find better solutions to filter out bad logos and recommend good ones. There are many existing logo generators, but they all have a common problem of being unable to differentiate between good and bad logos even though they can easily generate many. We want to incorporate a tool that would

Table 3: Test results of Prompt #3 – *simple minimal logo of a chicken, style of Pablo Picasso, (-) letters font*

	1	2	3	4	5
Prompt #3	Quality metrics				
Image	BRISQUE	CNNIQA	DBCNN	MUSIQ	NIMA
1	5.24	-2.35	56.89	60.31	4.84
2	18.83	-1.08	52.05	59.84	4.65
3	25.13	0.36	60.59	64.57	5.13
4	5.19	0.82	55.36	70.46	4.79
5	18.10	-0.78	52.03	59.16	5.17

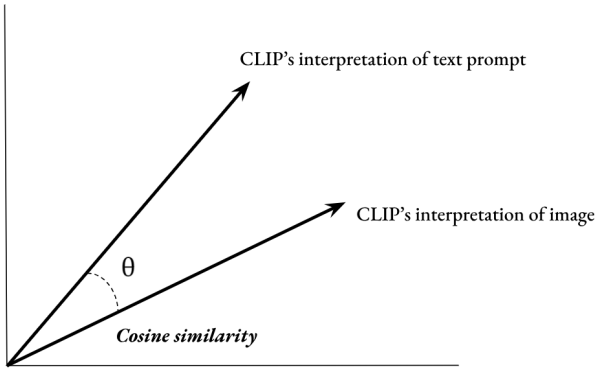


Fig. 4: An illustration of cosine similarity used in CLIP.

make the filtering process more efficient, otherwise we would have to be sifting through thousands of logos manually. It is well known that subjective evaluation is very time consuming and, consequently, error prone.

CLIP [11, 12, 29] is a model that is capable of projecting texts and images to a shared semantic space. It is multi-modal by combining natural language processing (NLP) and computer vision and zero-shot because it can be trained to carry out certain functions that it was not specifically programmed to perform. Its main usage is to predict the similarity between a text description and an image. The creators of CLIP formed a database of 400 million examples of both images and text in order to train the CLIP model's giant neural networks.

As an AI image generation model, the text encoder in Stable Diffusion is a particular kind of a transformer language, which employs the CLIP model (the second version of Stable Diffusion uses a variant of the CLIP model called OpenCLIP). This was discussed earlier in the section.

Stable Diffusion uses the text encoder only from the CLIP. As shown in Figure 4, the first step is tokenization, which is the way for a computer to understand words. The following step of embedding in Stable Diffusion is fixed by the CLIP model, i.e., learned through training. Proper embeddings were found to trigger arbitrary objects and styles, which would often lead to unimaginable images generated by the tool. The text transformer further processes the data and provides a mechanism to include multiple conditioning modalities, if existing, in addition to a text prompt. In Stable Diffusion, the embedding needs to be further processed by the text transformer prior to feeding into the process of predicting noise.

The CLIP itself also includes an image encoder. It is therefore straightforward to use CLIP to measure how similar a generated image is to its text prompt. Once the text encoder processes a text prompt to

a mathematical space, and the image encoder processes an image to the same space. A convenient way is then to calculate the similarity between the two vectors, or series of numbers, in a multidimensional space. Figure 4 shows a visual representation of cosine similarity.

3.3. Two-Stage logo refinement with CLIP

By using CLIP as image qulity metric, we designed a two-stage refinement to better meets the design requirements and preferences on layout and shapes. For each text prompt, the refinement includes the following steps:

1. Generate a set of 80∼100 images with Stable diffusion.
2. Use CLIP to compare the text prompt with each image (i.e., first round CLIP scores).
3. Sort the similarity scores from the highest to the lowest.
4. Choose the image of the highest score and extract the sketch/outline from the image.
5. Use the stretch as input to Stable Diffusion through ControlNet.
6. Generate the second set of 25 images.
7. Use CLIP to compare the text prompt with each image (i.e, second round CLIP scores).
8. Compare the CLIP scores of the two rounds.

Table 4: Comparison of CLIP scores for the results of Prompt #1 - *lion emblem in a square, style of Clash of Clans, game icon*

First round CLIP scores		**Second round CLIP scores**
Top 25 scores	Bottom 25 scores	All 25 scores
40.41	32.49	36.61
39.62	32.48	36.50
38.22	32.41	36.41
37.97	32.40	36.33
37.69	32.31	36.23
37.66	32.28	36.14
37.58	32.18	35.95
37.38	32.15	35.90
37.17	32.05	35.86
36.75	32.04	35.68
36.64	32.01	35.55
36.59	31.99	35.41
36.46	31.95	35.39
36.39	31.44	35.29
36.34	31.30	34.78
36.22	31.30	34.76
36.16	31.28	34.69
36.15	31.22	34.60
36.04	31.13	34.29
36.04	31.12	34.28
35.98	31.04	34.25
35.92	30.82	34.17
35.89	30.55	33.51
35.82	30.28	33.39
35.68	29.93	33.23
Overall average (100 total): 34.31		**Overall average (25 total): 35.17**

To test the effectiveness of our two-stage method on the text-to-image logo generation, we conducted three experiments with the three text prompts that were mentioned in Tables 4- 6. Throughout the following experiments with WebUI [30], we set the CFG (classifier free guidance) scale to 7.5 and the sampling method to DPM++ 2M Karras, which is in general a good choice for computational efficiency. The first task included creating and collecting a set of text prompts. The installation of WebUI also came with the support for command-line operation. This was very useful for batch processing with a large number of text prompts. We created Python scripts to generate logo images for 46 text prompts, which resulted in 230 images in total.

Table 5: Comparison of CLIP scores for the results of Prompt #2 - *vector graphic logo of frog, simple minimal, (-) realistic photo details*

First round CLIP scores		**Second round CLIP scores**
Top 25 scores	Bottom 25 scores	All 25 scores
38.68	30.70	36.61
38.18	30.68	36.33
38.14	30.64	36.20
38.04	30.60	35.71
37.41	29.78	35.68
37.08	29.76	35.51
36.56	29.65	35.44
36.23	29.65	35.01
36.20	29.32	34.80
36.00	29.30	34.54
35.89	28.53	34.34
35.79	28.04	34.30
35.78	27.72	34.24
35.62	27.43	34.23
35.43	26.90	34.22
35.36	26.68	33.80
35.14	26.40	33.65
35.12	26.20	33.25
35.09	26.10	33.05
35.05	25.63	33.04
35.04	25.42	32.77
34.94	25.35	31.82
34.91	24.68	31.72
34.87	24.36	29.83
34.80	22.98	29.19
Overall average (100 total): 32.12		**Overall average (25 total): 33.97**

Table 6: Comparison of CLIP scores for the results of Prompt #3 - *simple minimal logo of a chicken, style of Pablo Picasso, (-) letters font*

First round CLIP scores		**Second round CLIP scores**
Top 25 scores	Bottom 25 scores	All 25 scores
38.91	31.94	36.85
37.06	31.89	35.33
36.06	31.71	35.11
35.56	31.70	34.97
35.41	31.64	34.88
35.11	31.48	34.54
35.04	31.20	34.49
34.97	31.05	34.39
34.88	30.58	34.23
34.73	30.08	34.02
34.72	29.91	33.55
34.67	29.90	33.29
34.61	29.84	33.18
34.57	29.68	33.05
34.56	29.65	33.04
34.41	29.56	32.81
34.26	29.49	32.73
34.16	29.37	32.62
34.00	29.22	32.51
33.91	29.09	32.37
33.89	28.69	32.21
33.75	28.42	32.20
33.71	28.19	31.48
33.63	26.76	31.43
33.54	26.12	30.79
Overall average (80 total): 32.39		**Overall average (25 total): 33.44**

From evaluating the first round CLIP scores, we have found those to be more consistent with the human visual assessment than those scores reported by the image quality metrics in Section 3.2. Overall, it is a better chance that a lower score in CLIP similarity shows a bad logo image than a high score showing a good logo image. The observation is consistent across the results of the three text prompts. It therefore exhibits a higher confidence in filtering out the images with lower scores. This automated process of scoring and filtering can help to reduce the time and effort required in visually assessing a

Fig. 5: A batch of 25 images generated with the sketch shown on the left for Prompt #1.

large data set.

Steps 4)-7) above are designed to evaluate the contribution of using a sketch in logo generation by Stable Diffusion. ControlNet [9, 10] is a neural network structure to control the diffusion model with additional conditions. It is a solution to the problem of spatial consistency. The extra input conditions therefore create a scenario of multiple modalities, which the text transformer accommodates. This was described earlier in the section.

The test sketch is extracted from the image with the highest CLIP score in the first round. In the second round generation with ControlNet, a set of 25 images are rendered. Figures 5-7 present the sketch and the corresponding second round of images for the three prompts. Apparently, the second round of images showed fewer bad logos than in the first round. This is a significant improvement.

We have done a comparative study of the CLIP scores between the two rounds for each text prompt, which are collected at Step 2 and 7, respectively. Tables 4- 6 show the results of comparisons for the three prompts. The average CLIP score for the second round is consistently higher than the average CLIP score for the first round.

4. Color Harmony and Manipulation for Logo Images

In this section, we introduce the color harmony for the logo images of our method. We discuss the identification and manipulation of color harmonies by applying K-means clustering technique to extract the main colors from an image. Subsequently, we discuss the process of removing achromatic colors to derive a finalized palette of main colors. Using this palette, we create a circular color array that facilitates the identification of color harmonies present in the image. We further illustrate the process of empirically manipulating the color palette to achieve different color harmonies, which may be more pleasing or suitable for specific applications.

4.1. Color Harmony Identification and Manipulation

Color harmony [4–6] is a well researched topic in color theory. It was based on studies that combined art and science to determine what colors are pleasing to the human eye. It utilizes geometric relationships on the color wheel to distinguish compatible color combinations. There are six main color harmonies [4, 6].

Color harmonies are important for visual impressions of artwork. They can affect one's behavior and impression of an image through color psychology, a study of how colors affect human perception. Stable Diffusion does not currently have an efficient way to ensure color harmonies through prompts. Our

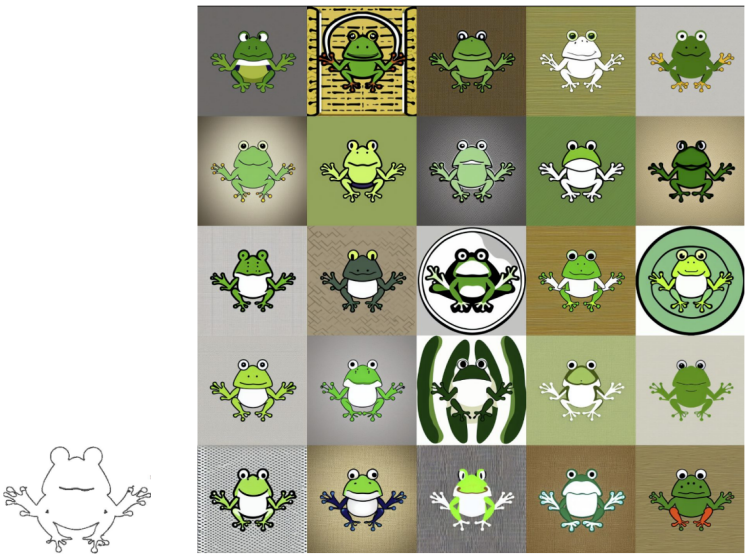


Fig. 6: A batch of 25 images generated with the sketch shown on the left for Prompt #2.

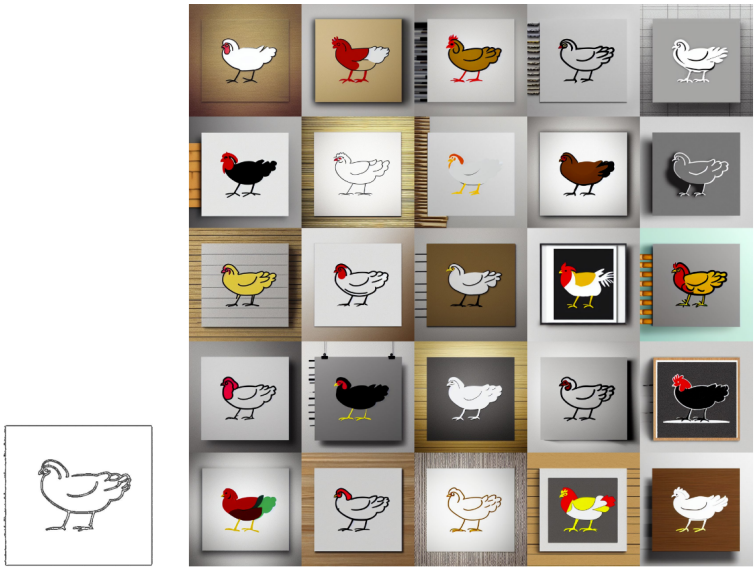


Fig. 7: A batch of 25 images generated with the sketch shown on the left for Prompt #3.

solution is to use a tool that generates a color palette for an image and classifies it under specific color harmonies. Then, we can identify a specific color and replace it with another to create a more satisfactory color harmony if desired. The palette generation uses K-means clustering to extract the main colors from an image and remove achromatic colors (black and white) from the palette. The extracted palette colors are then assigned to the color wheel, which is then used to classify the particular color harmony of that image.

K-means clustering [21, 22] is a popular machine-learning algorithm that is mainly used to analyze data clusters by grouping data points together to find patterns. It seeks out a fixed amount (k) of clusters, collections of data points that clump together due to similarities, in a data set. The specified number k represents the amount of centroids, imaginary or real site that portrays the center of a cluster, required in the data set. Consequently, every data point is assigned to a cluster while the algorithm tries to keep the centroids as small as possible.

The K-means algorithm starts with a selected group of centroids and then executes repetitive calculations to determine the location of centroids. The creation and optimization of the clusters stops when either the values of centroids are stable or the k number of replications have been completed.

4.2. Extraction of Main Colors

As described earlier, color harmony studies why certain combinations of colors together look more pleasing than others. Color harmony research has sought to discover models which predict pleasing color schemes. Hue template-based models are often used by artists and designers to help choose color schemes for their work, and have been applied in machine learning to classify images based on aesthetic quality [7] and to edit photos to improve their color [8].

Inspired by these ideas from color harmony, we investigate probabilistic models for studying color schemes, separate from their component hues. The study includes the following modules: A computational approach to clustering the main colors, or a palette, from an image. Removal of achromatic colors from the palette to finalize the main colors. Representing the main colors in a circular array to identify color harmonies that are present in the image. Empirical manipulation of color palette and replacement of colors for a different color harmony.

In this section, we describe the processes to extract the main colors from an image and classify those by color harmony. The computational approach of clustering the colors is designed to operate in the common RGB color space. Here, K-means clustering is used to generate a palette of primary colors in a generated logo image. The extraction process is to group similar colors in the RGB space and then use the centroids (i.e., mean points) to represent the color of each group.

It is worthwhile to note that brightness largely impacts the perception of colors and saturation. Colors that are almost black or white could cause problems in the following color harmony identification process. It is therefore more intuitively accurate if we remove colors relative close to black and white from the extracted palette based on the average chromatics of all the colors. The HSV (hue-saturation-value) color space [31] is used in this experiment. HSV is a good representation that models how colors appear, with attributes of hue and saturation.

The removal of achromatic colors from the extracted palette is first based on the following calculation of chroma values of the main colors. Note that, the ranges of hue, saturation and value in the HSV do not consider the requirements in saving the data to an 8-bit (i.e., [0, 255]) image format. In this paper, the HSV is an intermediate representation in the processes of color clustering, harmony identification and color manipulation. Hence, in the simulations, the conversion results are used without further scaling and conforming.

In the process of removing achromatic colors, let $i = 0, ..., L\text{-}1$ represent the i-th cluster in a total of L clustered colors. After the conversion from RGB to HSV, $Saturation[i]$ and $Value[i]$ denote the S and V, respectively, of the i-th cluster. For each cluster i, its chroma is first calculated as:

$$Chroma[i] = Saturation[i] * Value[i];$$

The mean and standard deviation of the chroma values are then computed as follows:

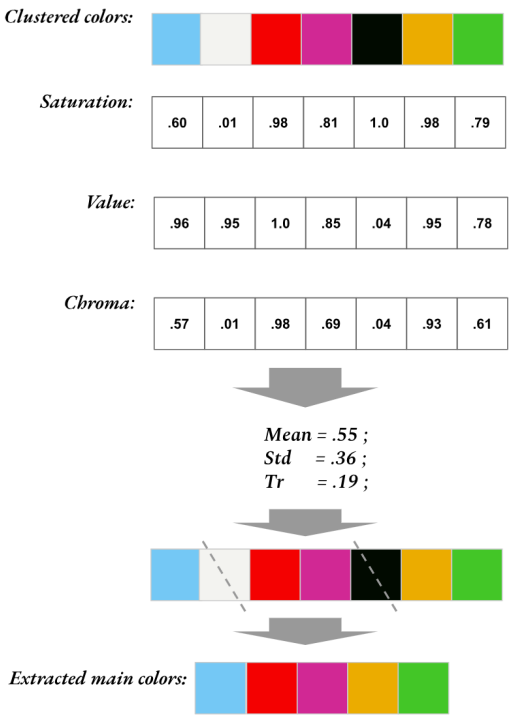


Fig. 8: Illustrative processes of removing achromatic colors from clustering. The final remaining colors represent the color scheme of the logo.

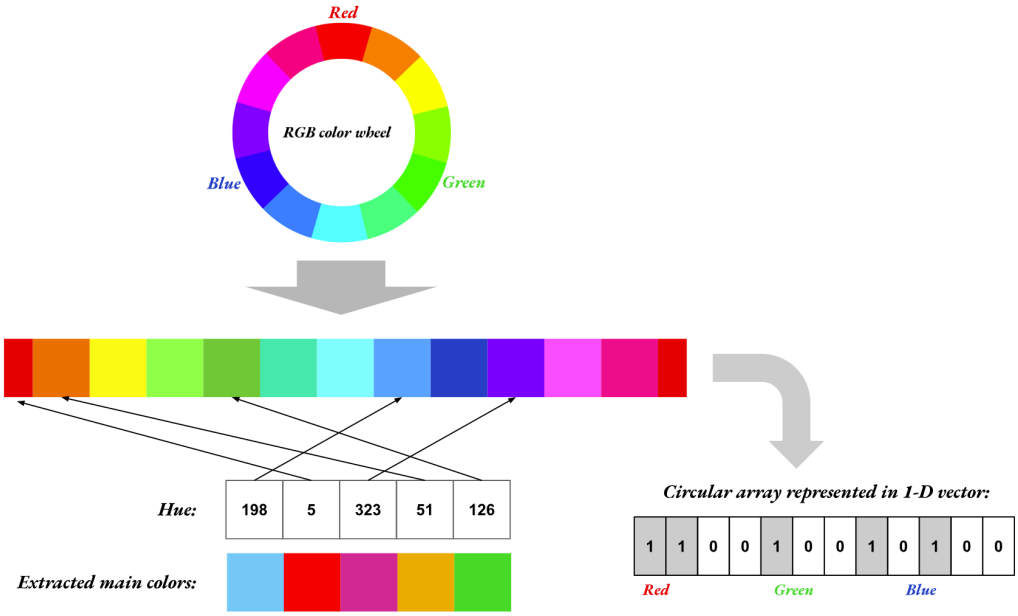


Fig. 9: Example process of creating a circular color array.

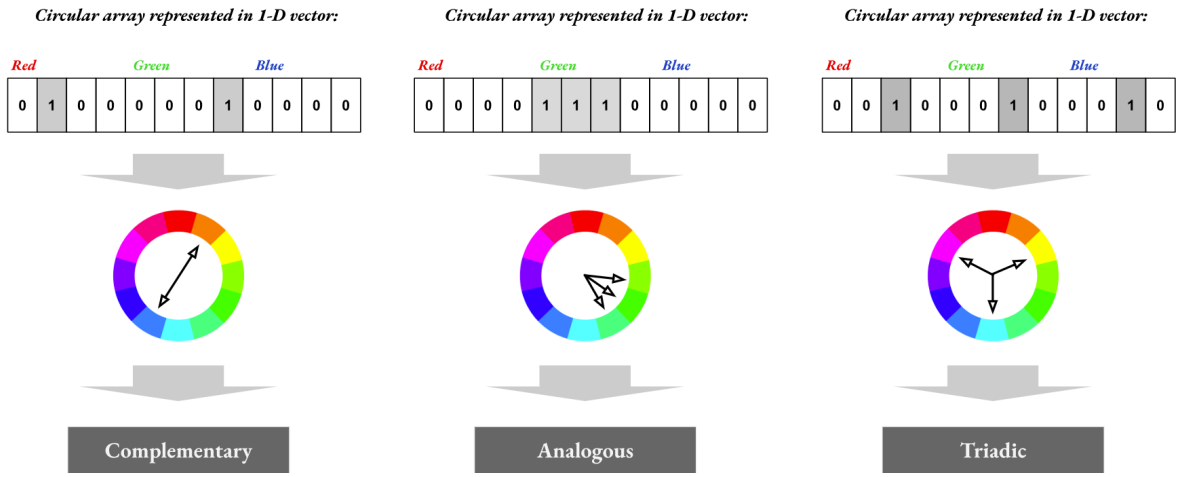


Fig. 10: Identifying color harmonies based on the circular color arrays.

$$Mean = \left(\sum_{i=0}^{L-1} Chroma[i] \right) / L$$

$$Std = \sqrt{\left[\sum_{i=0}^{L-1} (Chroma[i] - Mean)^2 \right] / L}$$

A threshold is thus determined for removing achromatic colors:

$$Tr = Mean - Std$$

For a clustered color i, if $Chroma[i] < (Tr)$, it will be marked as ineffective and finally removed from the extracted palette. An illustrative process of removing the achromatic colors is shown in Figure 8. Note that the qualification of achromatic colors from the clustering outcome depends on each logo. The removed colors are relatively close to black or white, which may cause problems in the color harmony identification. It can be observed in the example shown in Figure 8 that those colors may have small amounts of chromaticity (e.g., blue) detail mixed in.

4.3. Identification of Color Harmonies

After extracting a palette of main colors from an image, we can categorize them to the corresponding section on the RGB color wheel by their hues. In this paper, the color wheel is divided into 12 equally spaced slices. This is because color harmony definitions are usually based on spitting the color wheel into 3 primary, 3 secondary, and 6 tertiary colors [4, 6].

Then we specify the representative colors whose range includes our main color hues. By indicating the presence of a color in binary, we can produce a circular array that represents the main colors in the image. The process is illustrated in Figure 9. The circular arrays can then be used for identifying color harmonies through algorithmic computation.

In the process of identifying color harmonies, the underlying rules are to find the presence of nearby or opposite colors on the RGB wheel. The use of a circular array or 1-D vector can accommodate loops and modulo in the process of assessing the existence of a color harmony. A few example color harmonies are shown in Figure 10. One single image may exhibit multiple harmonies.

4.4. Empirical Manipulation of Color Harmonies in Logos

Adjustment of the colors presented in an image has been a common feature in professional photo editing tools such as Photoshop. In this section, we explore a new option of manipulating clustered colors to

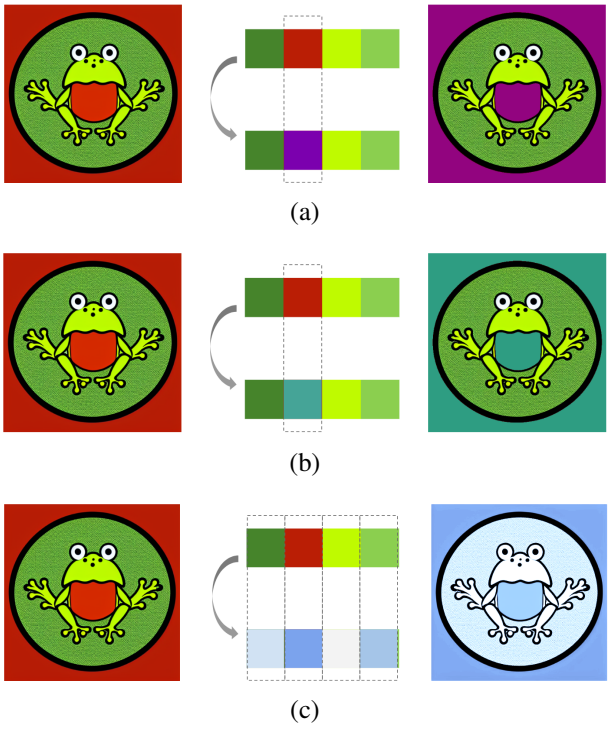


Fig. 11: Exemplary illustrations of (a)/(b) replacing a color in the palette, and (c) replacing multiple colors in the palette.

preserve a color harmony, or create a different color harmony, in a logo. This manipulation leverages the outcome of color clustering, and focuses on recoloring one or multiple areas selected based on the extracted main colors. As a result of color clustering, each pixel has been assigned to one of the clusters, represented by the centroids from the K-means clustering outcome. The process of replacing a color in an image is described in the following.

Let W and H denote the width and height of an image, respectively. Assume *cluster_r* is the index of a color in the palette that is to be replaced with a new color *color_t*. Note that a color representation includes the R, G and B values. At each pixel position *[x, y]*, where *x=0,...,W-1* and *y=0,...,H-1*, *pixel_cluster[x,y]* represents its cluster index. The color of the pixel at *[x, y]* will be replaced if (*pixel_cluster[x,y]* == *cluster_r*)

$$pixel\_color[x,y].r = color\_t.r;$$

$$pixel\_color[x,y].g = color\_t.g;$$

$$pixel\_color[x,y].b = color\_t.b;$$

4.5. Experiments and Results on Color Scheme Manipulation

Figure 11 shows an example of replacing multiple colors in a logo, which also results in a new image with analogous harmony. The choices of complementary and analogous colors can be obtained through online tools. In the above examples, we use the color calculator [32] to specify the original color and then the desired harmony. It then derives the target color or colors for replacement. The RGB values of the target color are consequently used in the processes described in Section 4.4.

Simply complementary colors are a great way to start exploring color harmonies and often appear quite bold and vibrant to the eye. The example shown in Figure 11(b) is to alter the opposite color of one of the green colors extracted from the logo. The resultant image still preserves a complementary, or split complementary, color scheme, but the overall impression is shifted, i.e., a little less lurid to some people.

Admittedly, it is a highly subjective process in evaluating a visual design. The perception of colors and choice of adjustment can vary from different creatives.

Analogous colors in combination with gray-scale (achromatic) colors can ensure the colors in a design do not clash while the combination is simple yet beautiful. In Figure 11(c), the logo is changed to a more analogous color harmony. By using this tool, users can alter the color scheme of an already satisfactory logo to better suit their preferences without having to generate a whole new logo.

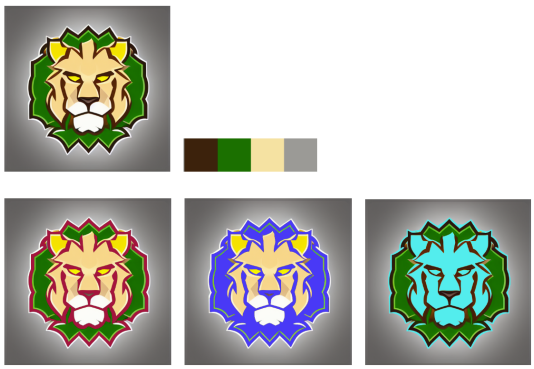


Fig. 12: (top row) A logo image of lion and its extracted Palette; (bottom row) examples of replacing one or more colors in the palette.

Fig. 13: (top row) A logo image of chicken and its extracted Palette; (bottom row) examples of replacing one or more colors in the palette.

Figures 12 and 13 present a few more examples of replacing one or more colors for an adjustment of color harmonies in a generated logo image. Note that, this does not require re-generating the logo through Stable Diffusion.

5. Conclusions and Discussions

In this paper, we conducted simulations for logo generation using WebUI, leveraging Stable Diffusion. The extraction of main colors from a logo was implemented in Python, and a tool was also developed in Python to manipulate the color palette for color replacement within the extracted palette.

Our initial experiments of using image quality metrics showed unsatisfactory results in measuring logo image quality. The main shortcoming of those metrics came from the inconsistency between their scores and our human visual assessment of the images. The metrics often gave higher scores to less visually appealing images and lower scores to more visually appealing images. This was apparently not useful in our goal of filtering inferior logo images and only recommending aesthetically pleasant designs.

We have then experimented with using the CLIP to compare the similarities between the output image

and the text prompt. The CLIP similarly scores showed a much better correlation with visual assessment. This automated evaluation can help to eliminate tedious subjective evaluation of a large image set and recommend visually pleasing logo images.

The use of ControlNet has proven useful by adding conditions to the logo generation. The results have exhibited better spatial consistency. The comparison of CLIP scores has also verified the improved performance when a sketch is added to the input in action to a text prompt.

The workflow consists of multiple modules, where each imposes a subproblem for optimization. Each of those subproblems may be worth further investigation to improve. An integration of the solutions that are presented in this paper have shown apparent potential to improve the efficiency of workflow.

The empirical manipulation of color scheme in the logo does not yet consider spatial coherence of pixels. It has limitations when dealing with e.g., colors in the areas of gradation and lighting variation. The approach proposed in [20] employed a more sophisticated decomposition to extract color palette from an image. A set of additive mixing layers, each corresponding to a palette applied with a varying weight. The recoloring is based on the geometry of the image. The subproblem of color re-harmonization in this section can also leverage the sophisticated multi-layer decomposition. This may be useful when a logo comes in gradation and change of lighting. This will be part of the future research which we continue improving on.

6. Acknowledgement

We would like to thank Professor Song-Hai Zhang and Dr. Ying-Tian Liu of Tsinghua University for their insightful advice that improves the quality of our study and the presentation of this paper.

References

1. Edwards, B., "Stunning - Midjourney update wows AI artists with camera-like feature," *Ars Technica*, June 2023.
2. Strickland, E., "DALL-E 2's Failures Are the Most Interesting Thing About It," *IEEE Spectrum*, August 2022.
3. Rombach, R., et al., "High-Resolution Image Synthesis with Latent Diffusion Models," *CVPR*, 2022.
4. O'Donovan, P., Agarwala, A. and Hertzmann, A., "Color compatibility from large datasets," *ACM SIGGRAPH*, 2011.
5. Shevell, S. K., *The Science of Color*, Elsevier, 2003.
6. Cohen-Or, D., et al., "Color harmonization," *ACM Trans. Graphics (TOG)*, vol.25, pp.624–630, 2006.
7. Nishiyama, M., et al., "Aesthetic quality classification of photographs based on color harmony," *CVPR*, 2011.
8. Lu, P., et al., "Image color harmony modeling through neighbored co-occurrence colors," *Neurocomputing*, vol.201, pp.82–91, 2016.
9. Zhang, L and Agrawala, M., "Adding conditional control to text-to-image diffusion models," *ArXiv*, 2023.
10. Zhao, S., et al., "Uni-ControlNet: All-in-One Control to Text-to-Image Diffusion Models," *ArXiv*, 2023.
11. Radford, A., et al., "Learning transferable visual models from natural language supervision," *ArXiv*, 2021.
12. Pan, X., et al., "Contrastive Language-Image Pre-Training with Knowledge Graphs," *NeurIPS*, 2022.
13. Goodfellow, I., et al., "Generative Adversarial Nets," *NeurIPS*, pp.2672-2680, 2014.
14. Arjovsky, M., et al., "Generative Adversarial Networks," *ICML*. pp.214-223, 2017.
15. Karras, T., et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation," *ICLR*, 2018.
16. Kang, M., et al., "Scaling up GANs for Text-to-Image Synthesis," *CVPR*, 2023.
17. Ho, J., Jain, A. and Abbeel, P., "Denoising Diffusion Probabilistic Models," *NeurIPS*. (2020)
18. Song, J., Meng, C. and Ermon, S., "Denoising Diffusion Implicit Models," *ICLR*, 2020.
19. Ramesh, A., et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents," *ArXiv*, 2022.
20. Tan, J., Echevarria, J. and Gingold, Y., "Efficient palette-based decomposition and recoloring of images via RGBXY-space geometry," *ACM Trans. Graphics (TOG)*, vol.37, 2018.
21. Celebi, M. E., "Improving the performance of k-means for color quantization," *Image and Vision Computing*, vol.29, 2011.

22. Zheng, X., et al., "Image segmentation based on adaptive k-means algorithm," *EURASIP Journal on Image and Video Processing*, 2018.
23. Podell, D., et al., "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis," *ArXiv*, 2023.
24. Mittal, A., Moorthy, A. K. and Bovik, A. C., "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Processing*, vol.21, no.12, 2012.
25. Kang, L., et al., "Convolutional neural networks for no-reference image quality assessment," *CVPR*, 2014.
26. Zhang, W., et al., "Blind image quality assessment using a deep bilinear convolutional neural network," *IEEE Trans. Circuits System Video Technology*, vol.30, no.1, 2020.
27. Ke, J., et al., "MUSIQ: multi-scale image quality transformer," *CVPR*, 2021.
28. Talebi, H. and Milanfar, P., "NIMA: neural image assessment," *IEEE Trans. Image Processing*, vol.27, no.8, 2018.
29. Wang, Z., et al., "CRIS: CLIP-Driven Referring Image Segmentation," *CVPR*, 2022.
30. AUTOMATIC1111, *Stable Diffusion WebUI.* https://github.com/AUTOMATIC1111/stable-diffusion-webui
31. Schwarz, M. W., et al., "An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models," *ACM Trans. Graphics (TOG)*, vol.6, no.2, 1987.
32. Quiller, S., *Color Choices: Making Color Sense out of Color Theory*, Watson-Guptill, 2002.