Image Reproduction

TNM097

Project 1.1

Erik Dahlström - Erida600
Filip Hamrelius - Filha243

Linköping Universitet

SE-581 83 Linköping

013-28 10 00, www.liu.se

Abstract

In this project, a database with smaller images has been utilized in a grid to replicate larger images. The database used contained 200 famous brand logos. The reproduced images had a final size of 2000 x 2000 pixels, which in turn was divided into a grid system with smaller images, each spanning 25 x 25 pixels each. The first reproduction pipeline uses the CIELAB colorspace and all the 200 images. With this method and dataset, the program can successfully reproduce a detailed image, while the smaller images remain distinguishable from each other. The program achieves this by examining the smaller regions within the grid system and calculating the Euclidean distance between the regions colorspace and potential replacements within the dataset. Images reproduced included everything between portraits, logos, dark images and paintings. Beyond the colorspace pipeline a second reproduction pipeline was also developed. This was based on the same concept as the first one but finds multiple candidates in the CIELAB colorspace. It then narrowed the selection down to the most suitable candidate by looking at the structural similarity between the region and the possible candidates. When both of these reproduction methods were developed, two different optimization methods were developed to reduce the dataset to fewer images. The first optimization method was based on the dataset and image similarity with regard to color, if more than one image had the similar color properties the rest were excluded from the reproduction, minimizing the size of the dataset. This optimization was implemented and discussed for two variations with different amounts of images. The second optimization method also concrates on minimizing the dataset, although with regard to the query image, i.e, the original image. This method finds the six most recurring colors in the original image and narrows down the dataset based on these colors and a threshold. Lastly the results were compared and discussed using three different measures of quality: Mean Squared Error (MSE), Signal to Noise Ratio (SNR) and S-CIELAB. The optimization methods were only explored together with the first reproduction pipeline based solely on the CIELAB color space.

After implementing the different methods and evaluating the quality measures, it can be concluded that a reproduction program capable of producing satisfactory results can be developed. Utilizing the mean of the Euclidean distance proved to be effective in representing perceived color differences within a perceptually uniform color space like CIELAB. Through optimization methods, the database used for reproduction can be significantly reduced. In other words, achieving results similar to a database with 200 images can be accomplished using a considerably smaller number of images. The second optimization technique, which involves examining the most dominant colors within the query image, yielded the most favorable results upon inspection with the naked eye, however the quality metrics were indistinguishable. Comparing all the results with regard to the quality measures it can be concluded that the quality metrics offer limited insights when comparing optimized and non-optimized images. It is only when compared with an image featuring a smaller grid that distinctions in measurements become apparent for the better. Thus, one can infer that these metrics might be more effective in assessing more extreme noise changes or when comparing solely to the original image rather than between different variations of reproduction. Further on, it can also be concluded that by using a more advanced reproduction that also considers structure similarity on top of the color space produces even more satisfactory results, especially in detailed areas such as gradient color differences and reflections.

1.Introduction

This chapter introduces the background and limitations about the project. The chapter also presents the fundamental assumptions made around the screen, reproduction resolution, database and the smaller image sizes.

1.1 Background & Aim

This project is the final project in the course Image Reproduction and Image Quality, TNM097. The project revolves around image reproduction using a diverse picture database. The aim is to generate a replicated image that closely mirrors the original from a distance, yet allows for clear identification of smaller details up close. The project's background includes basic color theory mostly regarding RGB, CIELAB and CIE XYZ. As well as quality measures such as, Signal To Noise Ratio (SNR), Mean Squared Error (MSE) and S-CIELAB. The project was developed in MATLAB and uses multiple built-in MATLAB functions. Besides this, the project uses an imported S-CIELAB function provided by the course from the author Xuemei Zhang.

1.2 Limitations

This report discusses two methods for reproduction and compares these two with three quality measures: SNR, MSE and S-CIELAB. The report then delves deeper and discusses the two optimization methods applied to only the reproduction in colorspace and tests these results with the same quality measures. No optimization on the second reproduction method is presented. To obtain comparable results the same image is used for all tests: The painting, *The Scream* by Edvard Munch this painting was represented as a *.jpg* with an original size of 964 × 1327 pixels. Only when necessary, references to other images are used.

1.3 Reproduction resolution

The choice was made to have a final image size of 2000 x 2000 pixels with the smaller images making up a grid spanning 25 x 25 pixels each. The screen used was a 2019 Macbook Pro 15 inch, meaning a 132 PPI, 1680 x 1050 screen resolution and a viewing distance of 20 inches, i.e 0.5 m.

1.3 Database

The database consists of 200 individual famous logos of brands. The database was constructed for this specific use case and the images were chosen to create a wide variety of different colors, structures and sizes. The database in its entirety can be seen in *Figure 1*.

Figure 1 - Database of 200 famous brand logos.

The database is represented in two ways in MATLAB: as a MAT-file with 200 indices, each containing L, A, and B channels, and as an RGB version. Both versions include images sized at 25 x 25 pixels.

2.Method

This chapter discusses the method and techniques used during this project. More specifically the reproduction pipelines, optimization techniques and the different quality measurements.

2.2 Reproduction on colorspace match

The colorspace reproduction pipeline takes an unknown query image regardless of size, resizes it to a predined measure of 2000 x 2000 pixels. It then extracts each channel: L, A and B to three different variables. The reproduction function then loops over the image 25 x 25 pixels at a time. Simultaneously it compares the current region to all the images in the dataset and calculates the mean of the Euclidean distance of each channel compared to the images in the database. Iteratively it finds the best image with the least distance in regard to the colors and finally replaces the region with this image. Lastly it converts the reproduced image back to the RGB colorspace.

Using CIELAB and the Euclidean distance, *equation 1*, was based on the fact that CIELAB is a perceptually uniform color space. I.e the distance between colors can be used as a metric for their perceived difference, allowing a numerical representation of color difference.The acceptability of this color difference depends largely on the application and the desired level of accuracy in reproduction. Generally, a color difference of three is commonly regarded as a satisfactory and accepted color match in numerous practical applications [1]. The threshold can therefore be set higher or lower to include more or less images, a threshold with three would narrow it down by a lot.

$$\Delta E_{ab} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}, \qquad (1)$$

2.3 Reproduction on colorspace match and structural similarity

The reproduction on structural similarity operates similarly to the reproduction on only colorspace. It uses the same pipeline to find potential matches within the colorspace. It narrows down matches with a threshold of 20 on the mean of the Euclidean distance. After this it uses the MATLAB function, *SSIM* (Structural similarity index for measuring image quality). The SSIM function serves as a metric for assessing the quality of an image by evaluating three characteristics: luminance, contrast, and structure. These are then multiplied together to produce a measure of image quality. This function takes two images in grayscale and returns an index between minus one, and one. The closer to the positive one, the better structural similarity the two images have [3]. The structural similarity pipeline uses the same grid system and compares the region of the query image to the potential matches found by the colorspace pipeline and decides the best match. It finally replaces the region with this image.

2.3 Optimization

2.3.1 Optimization on image individuality within database

This function was designed to optimize the dataset of images by filtering out images with similar color values. The optimization process involves several steps, it starts off by going through each image from the original dataset and first converts it from RGB to CIELAB color space. The function extracts the L, A, and B color channel values from the CIELAB images. The function then compares each image with existing images in the dataset to ensure they are sufficiently different. It does this by, again, calculating the mean of the Euclidean distance between the color values of the current image and those of existing images. If the mean difference is below a certain threshold, indicating similarity, the current image is not added to the dataset.

The user can from the main file choose the amount of images the database should contain: 0-200 as well as the threshold to filter on. Since the function revolves around a threshold it will aim to fill for example 100 images, but if the threshold is set too strict, this condition will not always be met. To avoid that the dataset changes size for each iteration the program first allocates 200 black images. In the end of the program the function makes sure that it removes any empty or black placeholder images from the dataset.

2.3.2 Optimization on query image

This function optimizes the database in regard to the query image. It uses the matlab function *rgb2ind* to create an indexed RGB image and retrieve the most recurring colors in the image. The optimization function then converts these colors to the CIELAB color space. It then uses the database with the LAB values to find the images that best match the colors retrieved based on a threshold using the mean of the Euclidean distance. Changing the threshold, does again, change the size of the dataset and how strictly similar the color has to be to get included.

The aim of this function is subsequently to narrow the dataset down and only use the most optimal images from the database for this specific query image. In the same way as the other optimization method, it uses basically the same loop and does also allocate for all 200 images and removes any excess in the end.

2.4 Quality measures

3.4.1 Signal to Noise Ratio - SNR

Signal to Noise Ratio (SNR) is a measure to compare signal to background noise. The summation is done over all pixels in the images, as can be seen in equation 2. The formula compares the original image to the noise of the second image; this is done by taking the difference between the original and the reproduced image. A higher SNR, expressed in decibels indicates a better signal quality, i.e less noise compared to the original image [1].

$$SNR = 10\log_{10}\left(\frac{\sum_{i,j}(g(i,j))^2}{\sum_{i,j}(g(i,j)-b(i,j))^2}\right). \tag{2}$$

3.4.2 Mean Squared Error - MSE

The Mean Square Error (MSE) calculates the squared difference between the reproduced image and original image. It is calculated using *equation 3*. Where *R(x,y)* is the original image, *S(x,y)* is the reproduced version. *M* and *N* are the image dimensions [2].

$$MSE = \frac{1}{MN}\sum_{y=0}^{M-1}\sum_{x=0}^{N-1}[R(x,y)-S(x,y)]^2 \tag{3}$$

3.4.3 S-CIELAB

S-CIELAB is an extension of the CIELAB colorspace designed to include spatial aspects and simulate the Human Visual System (HVS). The mean of the results obtained from S-CIELAB can serve as a measure of image quality, representing the average color difference between the original and reproduced image [2].

The parameters used within S-CIELAB:

- Pixels Per Inch: 132 PPI.
- Distance: 20 inches (0.5 m).
- Samples per degree: 46 samples.
- Whitepoint: [95.05, 100, 108.9] (CIE standard illuminant D65 in CIE XYZ color space).

3. Result

This chapter presents all the results found during this project, it includes various settings for both reproduction pipelines as well as different settings for both optimization methods. In the end, all quality measures on these results are also presented.

3.2 Reproduction on colorspace match

Settings:

- Final image: 2000 x 2000 pixels.
- Grid size: 25 x 25 pixels.
- Images used: 200.

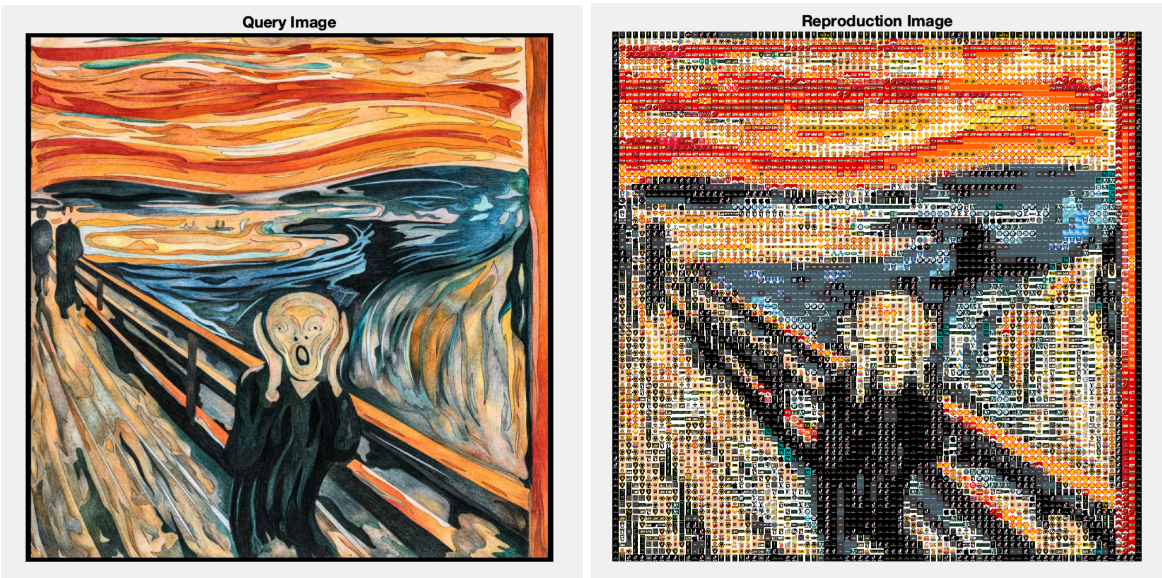

Figure 2 - Colorspace reproduction.

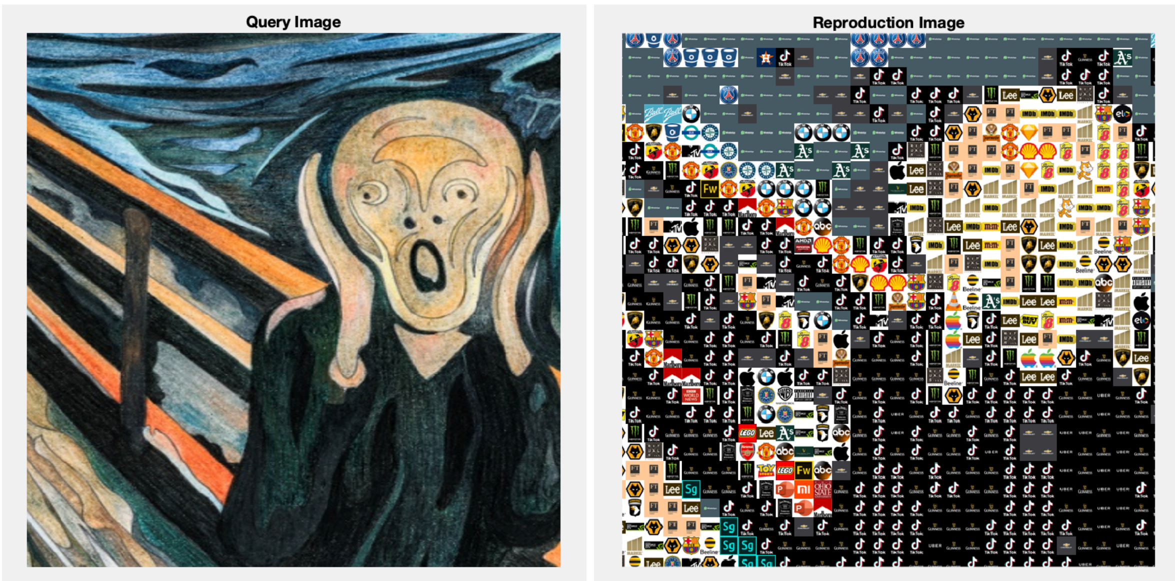

Figure 3 - Colorspace reproduction - zoomed in.

Settings:

- Final image: 2000 x 2000 pixels.
- Grid size: 10 x 10 pixels.
- Images used: 200.

Figure 4 - Colorspace reproduction - with smaller grid.

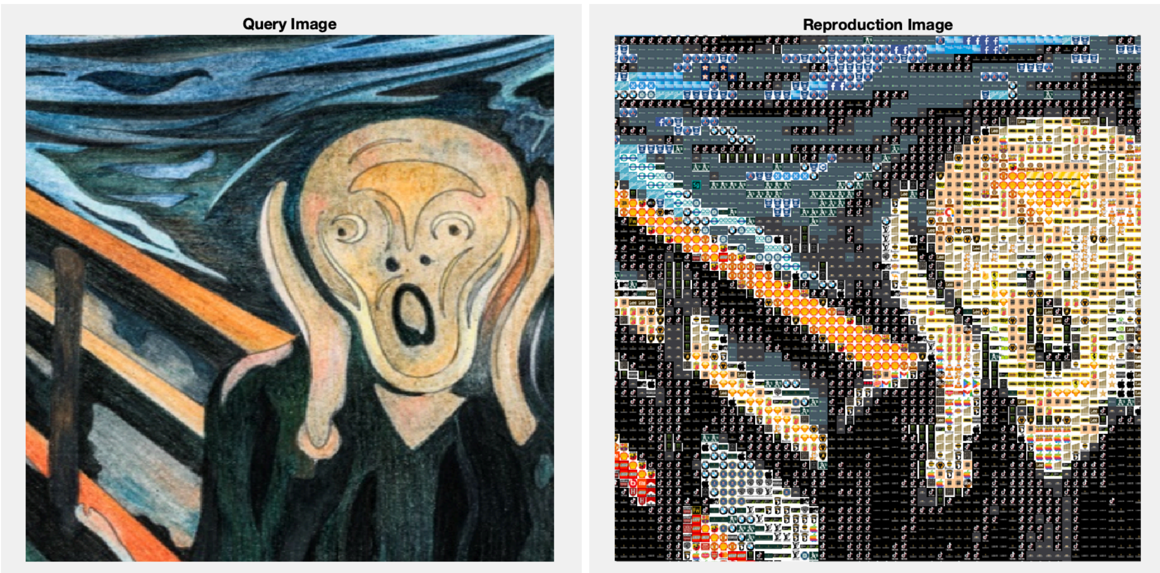

Figure 5 - Colorspace reproduction - with smaller grid, zoomed in.

3.3 Reproduction on colorspace match and structural similarity

Settings:

- Final image: 2000 x 2000 pixels.
- Grid size: 25 x 25 pixels.
- Images used: 200.
- Colorspace threshold before structural similarity comparison: 20

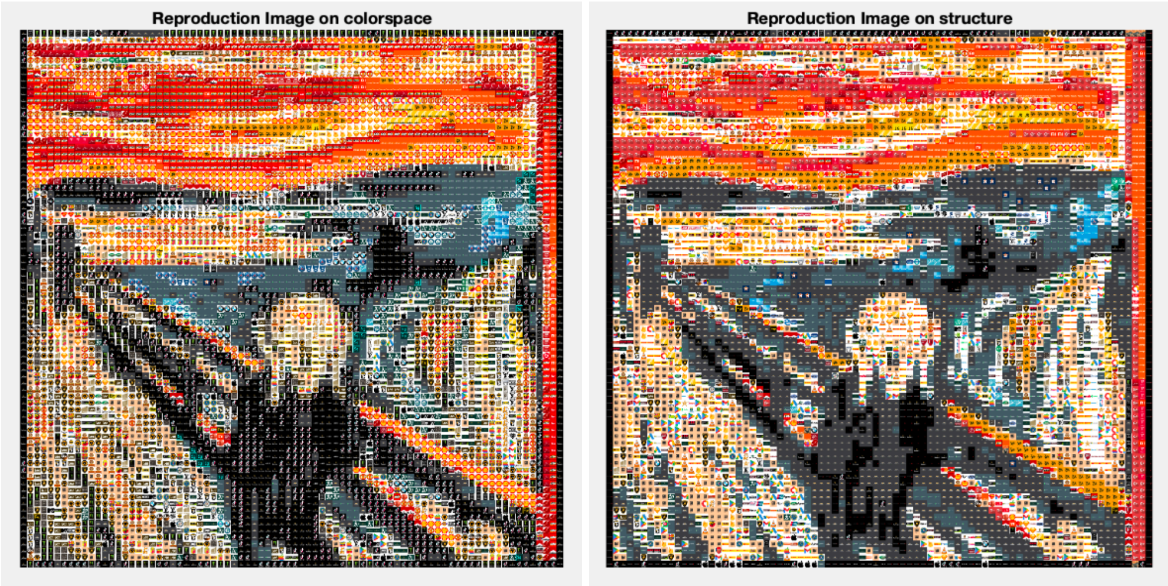

Figure 6: Colorspace vs Structural.

Figure 7: Query Image.

Settings:

- Final image: 2000 x 2000 pixels.
- Grid size: 25 x 25 pixels.
- Images used: 200.
- Colorspace threshold before structural similarity comparison: 20

Figure 8: Colorspace vs Structural: reference image.

Figure 9: Query Image.

3.4 Optimization

3.4.1 Optimization on database - Image individuality

Settings:

- Target amount of images: 50.
- Threshold: 30.
- Actual Images after filtration: 30.

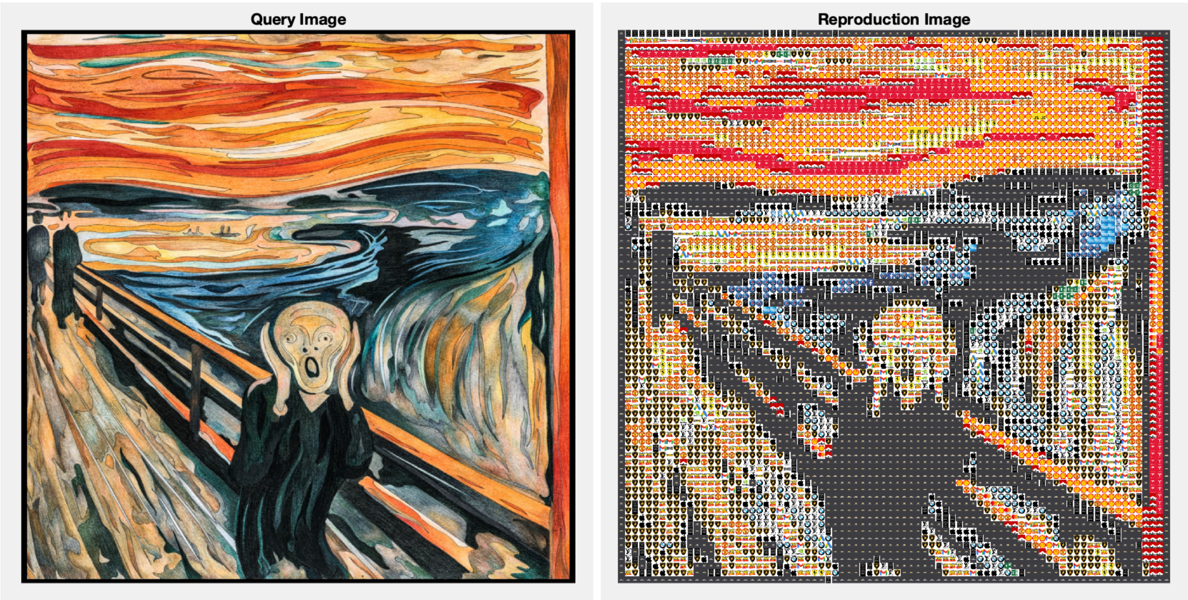

Figure 10 - Optimization on database - Image individuality: Target 50 images.

Figure 11 - Database after optimization: Target 50 images.

Settings:

- Target amount of images: 100.
- Threshold: 30.
- Actual Images after filtration: 66.

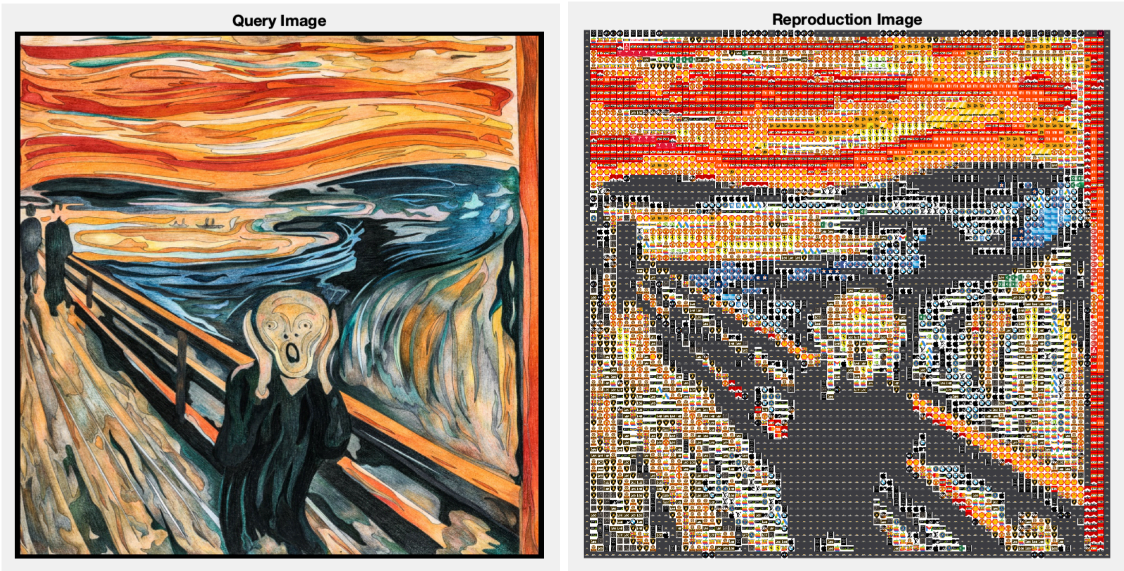

Figure 12 - Optimization on database - Image individuality - Target 100 images.

Figure 13 - Database after optimization: Target 100 images.

Settings:

- Target amount of images: 100.
- Threshold: 0.
- Actual Images after filtration: 100.

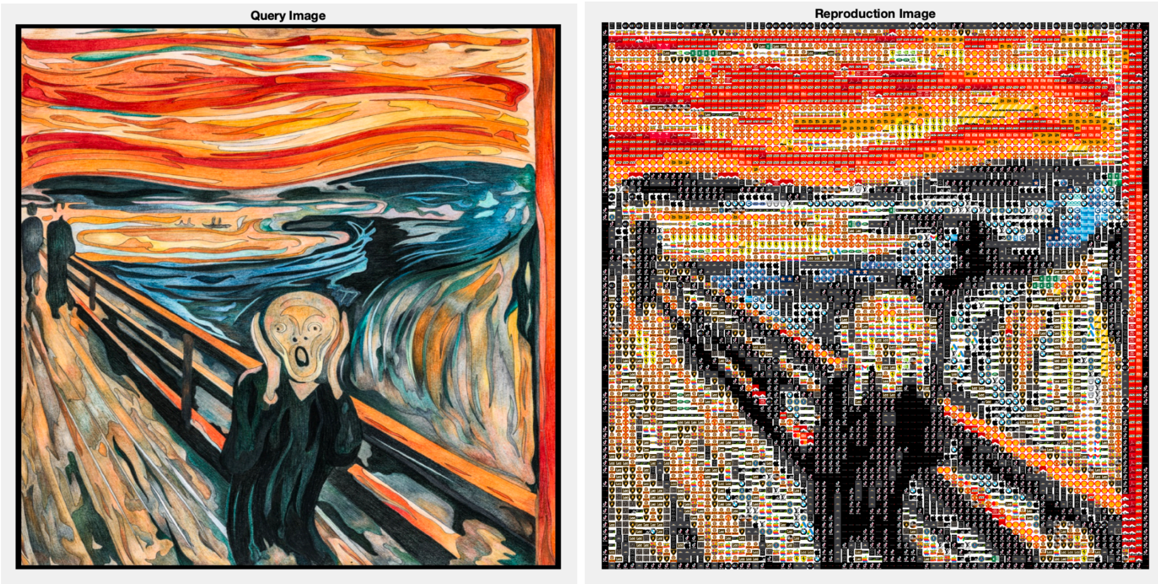

Figure 14 - Optimization on database - Image individuality - Target 100 images (0 Threshold).

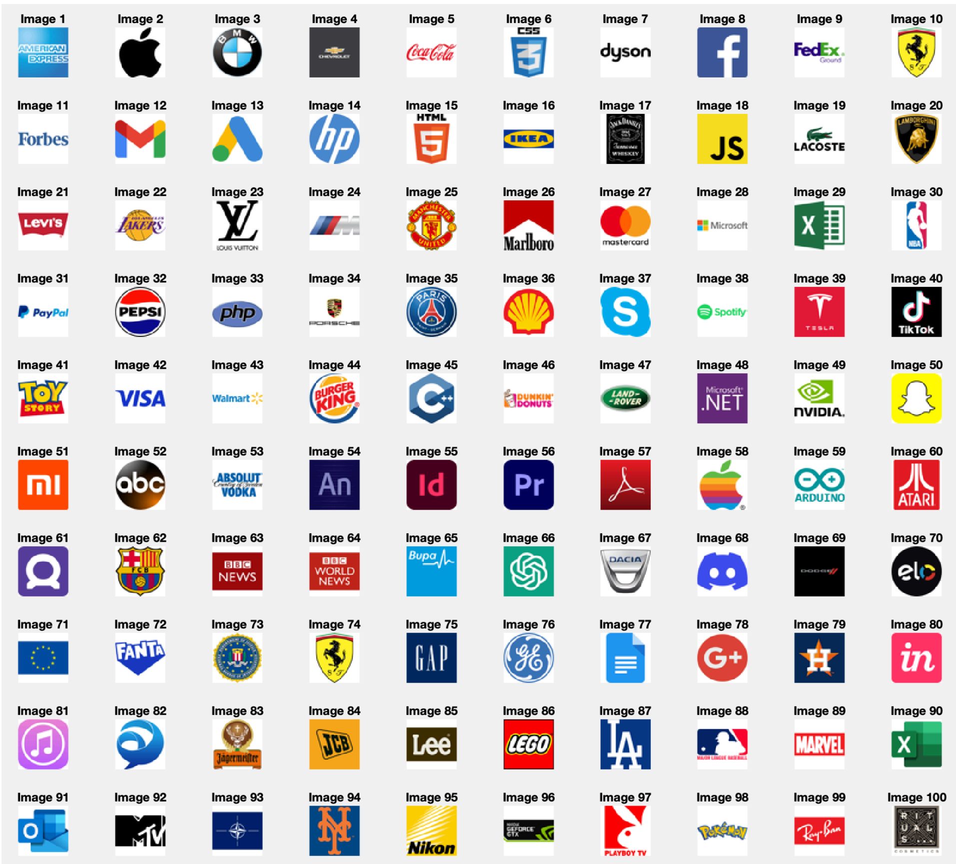

Figure 15 - Database after optimization: 100 images.

3.4.2 Optimization on query image

Figure 16 - Optimization - query Image with the 6 dominating RGB colors.

Figure 17 - The three most dominant RGB-values.

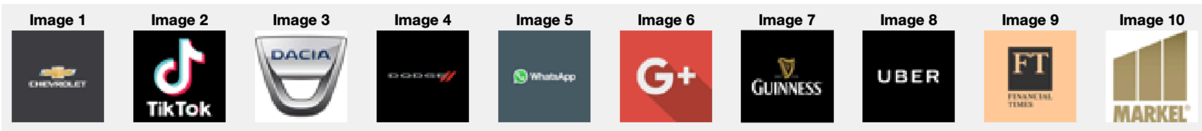

Figure 18 - Images closest to the 6 dominating colors: Threshold on mean of euclidean distance: 30.

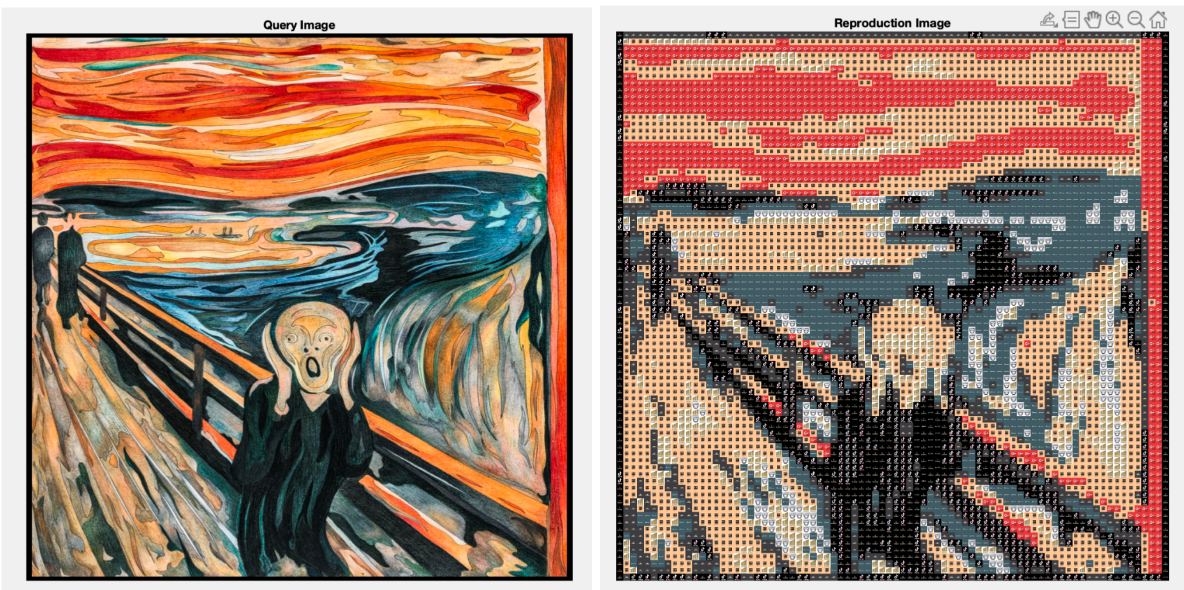

Figure 19 - Optimization on query image, 6 colors.

Figure 20 - Result image 6: Query Image with the 6 dominating RGB colors.

Figure 21 - Three most dominant RGB-values.

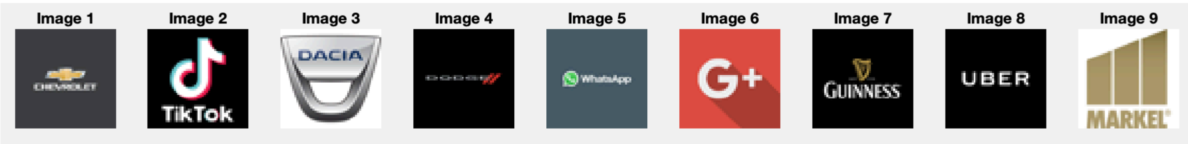

Figure 22 - Images closest to the 6 dominating colors: Threshold on mean of euclidean distance: 30.

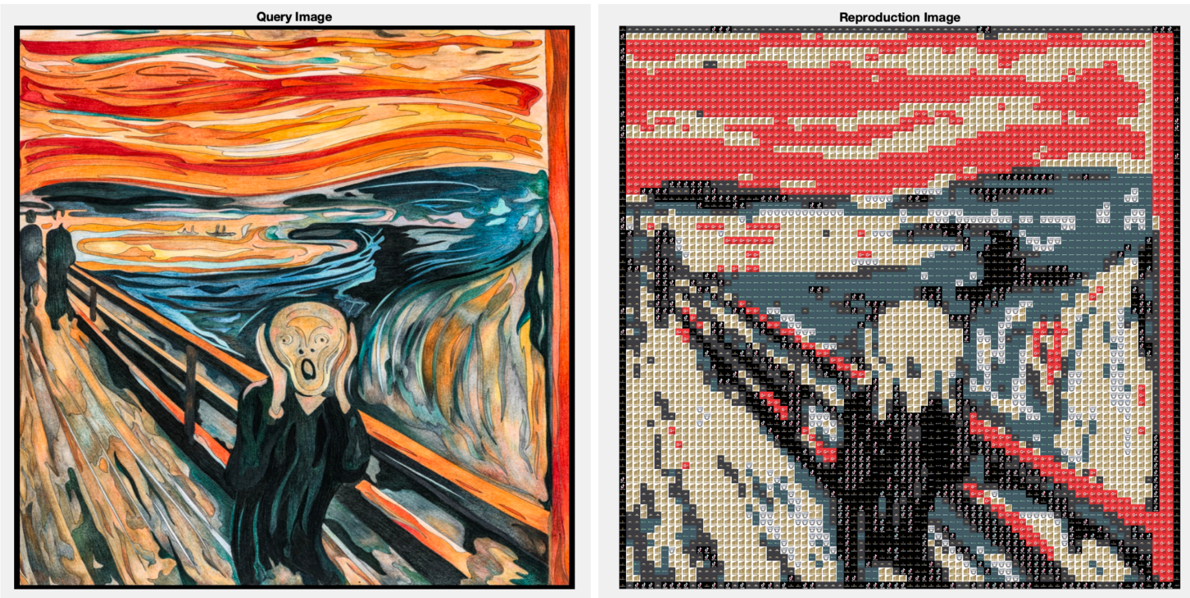

Figure 23 - Optimization on query image, 3 colors.

3.5 Quality measures

Quality measures for: SNR, MSE and S-CIELAB. These were calculated on the colorspace pipeline as well as the structural similarity pipeline with a final image of 2000 x 2000 pixels using a grid of 25 x 25 pixels and 200 images. Both optimization methods and their corresponding result for the first pipeline are also presented. For reference the quality was also calculated for a grid size of 10 x 10 pixels on the colorspace pipeline.

Table 1 - Reproduction on colorspace match

Settings:	**SNR**	**MSE**	**S-CIELAB**
Final Image: 2000 x 2000 pixels **Small Images:** 25 x 25 pixels **Images used:** 200 **Type:** Reproduction on colorspace **Opt**: No **Result:** *Figure 2*	3.50	0.12	2.79
Final Image: 2000 x 2000 pixels **Small Images:** 25 x 25 pixels **Images used:** 30 **Type:** Reproduction on colorspace **Opt:** On database similarity **Threshold:** 30 **Result:** *Figure 12*	3.66	0.12	2.8
Final Image: 2000 x 2000 pixels **Small Images:** 25 x 25 pixels **Images used:** 66 **Type:** Reproduction on colorspace **Opt:** On database similarity **Threshold:** 30 **Result:** *Figure 14*	3.82	0.12	2.7
Final Image: 2000 x 2000 pixels **Small Images:** 25 x 25 pixels **Images used:** 12 **Colors used:** 6 **Type:** Reproduction on colorspace **Opt:** On database with query image **Result:** *Figure 19*	6.046	0.071	2.14

Table 2 - Reproduction on colorspace match - Reference data

Settings:	SNR	MSE	S-CIELAB
Final Image: 2000 x 2000 pixels **Small Images:** 10 x 10 pixels **Images used:** 200 **Type:** Reproduction on colorspace **Opt:** No **Result:** *Figure 4*	5.35	0.084	1.87

Table 3 - Reproduction on colorspace match and structural similarity

Settings:	SNR	MSE	S-CIELAB
Final Image: 2000 x 2000 pixels **Small Images:** 25 x 25 pixels **Images used:** 200 **Type:** Reproduction on structural sim. **Opt:** No **Result:** *Figure 6*	5.59	0.079	2.39

4. Discussion

This chapter discusses the problems that were solved during the process as well as reasonings behind methods used and structure of functions. It also discusses the findings of using different quality measures.

4.1 Database

The use of brand logos contain a lot of dead space when represented as a *.jpg* since it needs a background. There are not a lot of logos that fill the entire space when encapsulated in a square. This was both positive and negative, the ones that did fill the entire square, *Figure 24*, creates an excellent filler for the final image since they mostly represent one single color. On the other side there are those logos that have a lot of empty space, *Figure 25*. These are great for representing white but do not bring any new colors to work with. It also creates, in some cases, weird artifacts on the reproduction image. An example of these kinds of artifacts can be seen in the face of *Figure 8* when looking at colorspace reproduction.

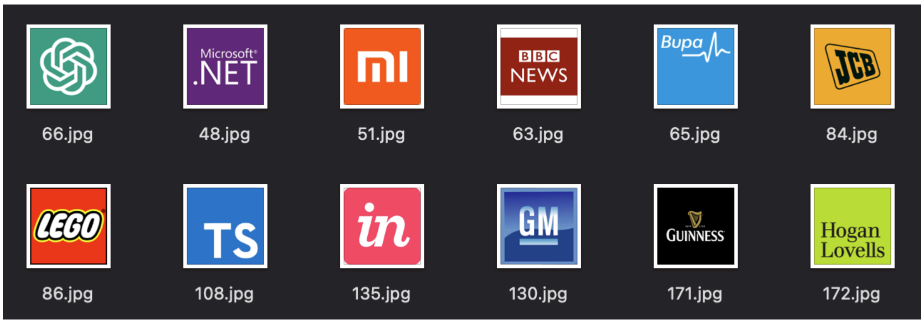

Figure 24 - Logos that cover the entire square.

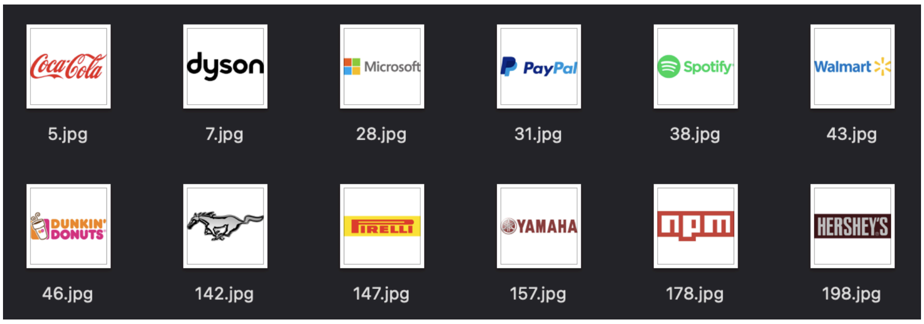

Figure 25 - Logos that do not cover the entire square.

4.2 Reproduction

As mentioned in the background, the query image was always immediately resized to be 2000 x 2000 pixels. The grid system was created with 25 x 25 pixels for the query image and 10 x 10 pixels for the reference image. These values were chosen because 25 and 10 are divisible by 2000, ensuring an even grid. Choosing other dimensions could result in a cut off on the smaller images or leaving original pixels on the query image. The decision to use 2000 x 2000 pixels for the query image was specifically made because it is a reasonable size for an image to be displayed on a modern high-resolution display. Since all resizing occurred at the beginning, immediately after reading the original image, the program ensures a consistent size throughout. This consistency guarantees that reproduction methods, optimization methods and various quality measures share the same starting point, producing tangible and comparable results. The downside of resizing to a specific resolution is that the original image gets cropped, particularly when using a rectangular resolution. Additionally, 2000 x 2000 pixels is a relatively high resolution, and many of the images used in the project fell well below this resolution. This implies that they were upscaled before being processed through the pipelines, which could potentially have affected the results negatively. However, since the same image and resolution were consistently used this issue does not impact the results presented in this report.

In *Figure 2-5* the result of the colorspace reproduction is shown. It can be concluded that the reproduction works very well and that the small images are still distinguishable. The size of the smaller images as well as the final image were chosen by trial and error. *Figure 4-5* shows the effects of having a smaller grid, i.e a higher resolution. This does increase the detail in the reproduction, but the quality and detail of the smaller images are lost. In theory the reproduced image could be built by so small images that they would look like normal pixels, this would result in an impressive representation of the original image, but defeat the purpose of the reproduction.

The reason why the CIELAB color space was used for the reproduction pipeline was due to perceptual uniformity. This property makes it well-suited for accurately representing color differences. Using this colorspace and the mean of the Euclidean distance results in a simple and effective method for comparing color differences. Although it would be interesting to experiment with alternative methods for comparing and calculating color differences. The current method is rather rudimentary and ultimately relies on the mean value, providing a broad representation of each image and region.

Reproduction on colorspace match and the structural similarity reproduction results, at first glance in a quite similar output image. However, from the quality metrics values that can be seen in *Table 2,* the conclusion can be drawn that this function results in a better visualization overall. Choosing images from the dataset based on color and structural similarity results in images that potentially reduce noise level in the reproduced image, which results in higher SNR-values. Beyond this, the structural similarity increases the chance that the image is more relevant for the specific region from the original image. Something that leads to an accurate reproduction of the area thus lowering the MSE value. This can also be seen with the naked eye in *Figure 6 - 9*, where *Figure 8* shows the effectiveness of the correct structure, shining through when representing reflection and gradient color changes.

4.3 Optimization

4.3.1 Optimization on database

In the Optimize database function, a second database is created with empty, black LAB color channels, by pre-allocating the dataset cell array it does not need to increase in size during the function loop. This step is done since the method used can produce less images in the dataset than the user requested. If one were to choose a strict threshold and a lot of images, there will be less images added than requested since the logos are too similar. The last step in this optimization function is therefore to remove any black images still remaining in the dataset. This could also be done using an empty dataset and then adding the images during the loop, although this does change the size for every iteration of the loop. The implemented version only changes the size of the dataset once, when the iterations have been completed.

As can be seen in the result *Table 1,* the optimization method does rarely provide the user with the correct number of images. In all cases the function provided less images than requested. This is because all the images in the dataset are very similar, they have similar structure and a lot of them even have similar color. Meaning, to find a balance between the amount of images and a threshold to filter on became quite difficult. Having a lower threshold meant including more images and a higher threshold excluding more images. There is no middle ground because of the nature of the dataset. Using a low threshold essentially provides the user with the first images in the original dataset, without any filter or optimization, this can be seen by comparing *Figure 12* and the database result in *Figure 13* with *Figure 14* and database result in *Figure 15.* Having a stricter threshold provides a well optimized database but with fewer images than asked for, this can be seen in *Figure 10 - 11*. Having a dataset with more variety would solve this problem.

4.3.2 Optimization on query image

By using *rgb2ind* the program can distinguish and retrieve the most recurring colors from the original image. Scenarios where the dominant colors might fail as an optimization method is when the function uses too few colors as dominant colors. The reason why the function uses 6 recurring colors is because the reproduction images started losing details with less colors. It was tested with 3,4 and 5 colors. It was found that a lot of details often have a different color from what is dominating, a value of 6 therefore became a good middle ground for the images used during this project. The differences between using 3 and 6 colors can be seen in the result *Figure 16 - 23.* As can be seen in *Figure 23*, a reconstructed image will still be achieved, however, it will be produced and displayed in a slightly inaccurate way, losing a significant amount of detail when some of the minority colors are not included. On the other hand one does not want to use too many dominating colors, if for example 10-20 colors would be used, all the colors visible would be included and the optimization would lose its intention.

4.4 Quality measures

4.4.1 Subjective discussion

The data gathered from all the quality measures generally follow a reasonable pattern. Increasing the resolution, i.e, making the grid smaller or enlarging the query image, results in a higher SNR value, indicating less noise. The S-CIELAB value decreases, and so does the MSE. Utilizing a more advanced method, which includes structural similarity, also yields better values across all quality measures, indicating reduced noise and increased similarity when passed through HSV within the S-CIELAB. This outcome is reasonable since the advanced method is based on the same pipeline as the one using only CIELAB, but it narrows down the color space and selects the most structurally similar image. Given the amount of similar images and colors within the dataset, this approach identifies numerous candidates. Choosing based on structure becomes crucial due to an overflow of good candidates concerning color. However, with a more diverse dataset containing fewer color duplicates, a different result might be obtained, potentially yielding better results with a color-only method.

The first optimization method, which narrows down the dataset based on image similarity within itself, produces results almost identical to the original database. This is reasonable because the dataset is quite repetitive. Removing similar images essentially creates the same result, even though the dataset is halved or more. The outcome also depends on the number of images available for that specific query image. If an image with very specific and uncommon colors undergoes the same optimization, the database might perform worse, requiring all available colors in the dataset. A bit surprising was the quality measure results yielded by the second optimization technique. This optimization method outperformed everything else. This could be because it uses less images creating more repetition and less noise but still remaining in the exact colorspace.

The SNR measure was chosen for its focus on noise, as the reproduction is based on other images. It is interesting to examine how much noise is introduced when incorporating new characters, colors, and structures into the original canvas structure. The MSE measure was selected to provide additional insight into SNR. If both measures correlate, the result appears more tangible, functioning almost like a safety check. The S-CIELAB was mandatory for this project but offers better insight into how the human eye would interpret the image, presenting a different quality measure than SNR and MSE. In retrospect, it would have been interesting to consider SSIM as a strict quality measure as well. However, reasonably, the structural similarity pipeline would perform better than the strict CIELAB since the method is based on the same function: SSIM.

4.4.1 Objective discussion

In reference to the ppi value and viewing distance used for the quality measure S-CIELAB. The PPI dimensions were based on our computer specifications and the viewing distance was decided through iterative testing. Drawing from insights gained in a prior laboratory session, specifically Lab 3, where a PPI value of 132 and a viewing distance of 20 inches (equivalent to 0.5 meters) were employed. These values worked well for the lab results and agreed well with what was perceived in reality. Therefore, the same values were used again.

As can be seen in *Table 1*, the quality measures do not differentiate a lot between the optimized and non optimized images when looking at the first optimization method. Since the results differ the same amount on all measures the conclusion can be drawn that the database contains a lot of similar images, and for this specific input image, all of them are not used and the optimization does not make a lot of difference. Although when optimizing on the most recurring colors in the image and only choosing images that align with these, a better result is given. This is probably because single images that would find their way into the non optimized image in specific regions are removed in its entirety and more of the same image is used, reducing noise and providing better result values, especially for the SNR and MSE.

The quality measures do not give a lot of insight between the optimized and not optimized images for the first optimization method. Only when compared to an image with a smaller grid, the other reproduction pipeline and more intrusive optimization as method 2, does the measurements begin to differentiate significantly for the better. This can be seen by comparing *Table 1-3*. The conclusion can therefore be drawn that the measurement might be better suited for more extreme noise changes. As discussed earlier, concerning these results, it would have been interesting to apply additional quality measures that consider different criteria, such as SSIM with structural similarity.

5. Summary

This project demonstrates the feasibility of utilizing smaller images from a database to replicate bigger images effectively, relying only on color differences but also on both color and structural similarity. By using reproduction pipelines based on displaying an image around 2000 x 2000 pixels, all of the logos from the reproduced image could be identified. Colorspace matching along with optimization techniques and structural similarity were constructed in a way so that the dataset size could be reduced from 200 images down to 10 images and still cover the most of the color space for the reproduced image in a sufficient way. Although this is subjective, the images can in all cases be clearly identified.

Even though many good reproductions were achieved, there are many approaches that still could be explored to reproduce the original image. Combining the optimization techniques with the structural similarity pipeline, exploring other measurements for color differences and potentially optimize the program as a whole.

6. Literature

[1] Sasaan Goran, January 9, 2018, Digital Halftoning and Color Reproduction, Course book - TNM059, Department of Science and Technology Linköping University.

[2] Marius Pedersen and Jon Yngve Hardeberg, Gjøvik 2009, Survey of full reference image quality metrics, Høgskolen i Gjøviks rapportserie, 2009 nr. 5.

[3] SSIM, MathWorks, https://se.mathworks.com/help/images/ref/ssim.html, Collected: 2024-02-26.