Technische Universität Berlin

Faculty of Electrical Engineering and Computer Science

Institute of Computer Engineering and Microelectronics

Computational Psychology

An Evaluation of Content-Adaptive Subsampling for Image Compression

A thesis submitted for the degree of

Bachelor of Science in Media Technology

by

Eduardo Luiz Rhein

Matriculation Number: 402800

First examiner: Prof. Dr. Marianne Maertens

Second examiner: Prof. Dr. Guillermo Gallego

Submitted on March 16, 2023

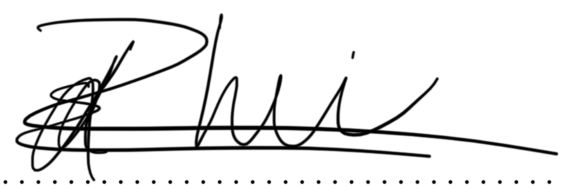
Statutory Declaration

Eidesstattliche Erklärung

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed in this document.

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, 16.03.2023



Eduardo Luiz Rhein

Acknowledgment

First and foremost, I would like to express my sincere gratitude to my supervisor and head of the computational psychology department, Prof. Dr. Marianne Maertens, for her invaluable patience, continuous support and always being available when I needed guidance. I would also like to extend my sincere thanks to the professor of robotic interactive perception, Prof. Dr. Guillermo Gallego, for accepting my invitation to be my second examiner.

Finally, I would like to express my heartfelt appreciation to my parents, Alexandre and Debora Rhein, and my brothers, Eric and Arthur Rhein, for their unwavering support and encouragement throughout my years of study. This achievement would not have been possible without their love and support.

Abstract

Existing lossy image compression methods, such as JPEG, often use the chroma subsampling technique to reduce the amount of information stored in an image file by encoding the color information of images at a lower resolution than the brightness information, exploiting the fact that the human visual system has lower visual acuity for color changes than for changes in brightness.

In this thesis, I thoroughly analyse, implement, and evaluate an algorithm that adaptively subsamples the luminance component of images, in addition to just uniformly subsampling the chroma components. To determine whether the luminance component can be subsampled, this algorithm takes into account certain content characteristics in the images. To test the effectiveness of the algorithm, I performed several subjective image quality tests and I also analyzed the compression factor increase of compressed images using the content-adaptive subsampling method.

The results of the study suggest that the tested algorithm can further increase the compression factor of the images. However, the degree to which the compression factor increases significantly depends on the type of image and its content characteristics. When using subsampling schemes with low compression ratios, it can be difficult to notice any difference in visual quality between the uniform chroma subsampling method and the content-adaptive subsampling algorithm. However, for subsampling schemes with a higher compression ratio, the perceived visual quality of images compressed with the content-adaptive subsampling algorithm can significantly degrade and be worse than images compressed with the uniform chroma subsampling method.

Zusammenfassung

Verlustbehaftete Bildkompressionsmethoden wie JPEG, verwenden häufig das Chroma Subsampling Verfahren, um die Datenmenge von Bildern zu reduzieren, indem die Chrominanz im Vergleich zur Luminanz mit einer reduzierten Abtastrate gespeichert wird. Dabei wird ausgenutzt, dass das menschliche Sehsystem für Farbänderungen eine geringere Sehschärfe hat als für Helligkeitsänderungen.

In dieser Arbeit analysiere, implementiere und evaluiere ich einen Algorithmus, der zusätzlich zur Farbunterabtastung, die Luminanzkomponente von Bildern adaptiv unterabtastet. Um zu bestimmen, ob die Luminanzkomponente unterabgetastet werden kann, berücksichtigt dieser Algorithmus bestimmte Inhaltseigenschaften in den Bildern. Um die Wirksamkeit des Algorithmus zu testen, habe ich subjektive Bildqualitätstests durchgeführt und ich habe auch die Erhöhung des Kompressionsfaktor komprimierter Bilder mit diesem Algorithmus analysiert.

Die Ergebnisse deuten darauf hin, dass der getestete Algorithmus den Kompressionsfaktor der Bilder weiter erhöhen kann. Der Kompressionsfaktor hängt jedoch sehr von den Inhaltseigenschaften ab. Bei niedrigen Komprimierungsverhältnissen kann es schwierig sein, einen Unterschied in der Qualität zwischen dem Uniform Chroma Subsampling Verfahren und dem inhaltsadaptiven Subsampling Algorithmus zu erkennen. Bei einem höheren Komprimierungsverhältnis kann sich jedoch die wahrgenommene visuelle Qualität von Bildern, die mit dem adaptiven Subsampling Algorithmus komprimiert wurden, erheblich verschlechtern und schlechter sein als Bilder, die mit dem Uniform Chroma Subsampling Verfahren komprimiert wurden.

Contents

1 Introduction

By the end of the year 2023, the volume of data generated, captured, duplicated and utilized globally is expected to exceed 100 zettabytes (one zettabyte is equal to one billion terabytes), a nearly tenfold increase from 2014, as illustrated in [12]. With the limited and very costly memory capacity for digital information that we have today, it is necessary to find efficient ways to reduce data size and the process of reducing the size of digital information is known as data compression [2] [9].

Data compression can significantly reduce the size of digital media, such as digital images, videos and audio. Reducing data size is very beneficial for several reasons, including saving storage space and reducing the bandwidth required for data transmission. Especially in today's digital age, where digital media are widely used in various applications and are increasing in size, effective data compression methods have never been more important.

Although reducing the size of digital information is essential, preserving its quality, be it visual or auditory, is also important, making it necessary to find a balance between achieving smaller file sizes and maintaining data integrity, as there is a limit that data compression can reach where the size reduction does not significantly affect the perceived quality.

One very common technique used in many compression formats for digital images and videos, such as JPEG [13] [15] and MPEG [5], is the so-called chroma subsampling technique, which can reduce the size of image and video files without significantly changing the perceived quality.

1.1 Compression Methods for Images

As described in [2] and [9], data compression methods can be divided into two types: lossless and lossy compression methods. The main difference between lossless and lossy compression methods is the amount of information that is lost during the encoding process. When using lossless compression methods, the perceived quality of the decompressed data is identical to the original data. In other words, lossless data compression methods can reduce data size while still preserving the quality of the original data. Since lossy compression methods can achieve higher compression factors than this type of compression, lossless compression methods are typically used when high compression factors are not required or when it is important to preserve the quality of the original data in the uncompressed file. An example where the lossless compression method is preferred is when working with text files or executable codes, as it may be necessary to compress these files without any loss of information. Among others, a widely used lossless compression algorithm is the Huffman coding algorithm, a compression method that uses the probabilities of occurrence of symbols in the data set to be compressed to determine variable size codes for each symbol. This algorithm was developed and published by David A. Huffman in 1952 [3].

On the other hand, there is information loss during the encoding process of lossy compression methods. The advantage of lossy compression methods over lossless compression methods is the ability to achieve higher compression factors. However, since some information will be lost, depending on the desired compression factor, the perceived quality of the decompressed data may not be as good as the original data. Lossy compression methods are widely used for digital media, such as digital image and audio, because they can take advantage of the way the human visual and auditory systems perceive information and as result, some information can be removed without significantly impacting the overall perceived quality.

The calculation of the compression factor requires the size of the input data and the size of the output data and can be calculated using the following equation:

$$\text{Compression factor} = \frac{\text{size of the input data}}{\text{size of the output data}}$$

If the result of the equation is greater than 1, it means that the size of the output data is smaller than the input data. In other words, it indicates compression. If the result of the equation is smaller than 1, then it indicates an expansion in size.

JPEG, a lossy compression method introduced in 1992 by the Joint Photographic Experts Group, is a commonly used lossy compression algorithm for digital images. The algorithm consists of several steps [13] [15] [11], among which the most important are: chroma subsampling, DCT (discrete cosine transform), quantization and entropy encoding. With JPEG, it is possible to choose how much one wants to compress the image when encoding. Depending on the compression rate set, the chroma subsampling in JPEG is not carried out at all or with a halving or quartering of the color channels.

The DCT is used to convert the image's information to a frequency domain representation so that a quantization matrix can be used to filter high frequencies, which are less noticeable to the human eye. Also, depending on the chosen quality level, which directly influences the elements of the quantization matrix, one can control the level at which one wants to reduce the amount of information in the high frequency components, and thus further increase or decrease the compression factor. And finally, the last step is the entropy encoding, a lossless process, which takes the quantized DCT coefficients and maps them to a new set of values that are more efficient to represent. The main steps of the JPEG compression encoding process are illustrated in figure 1.

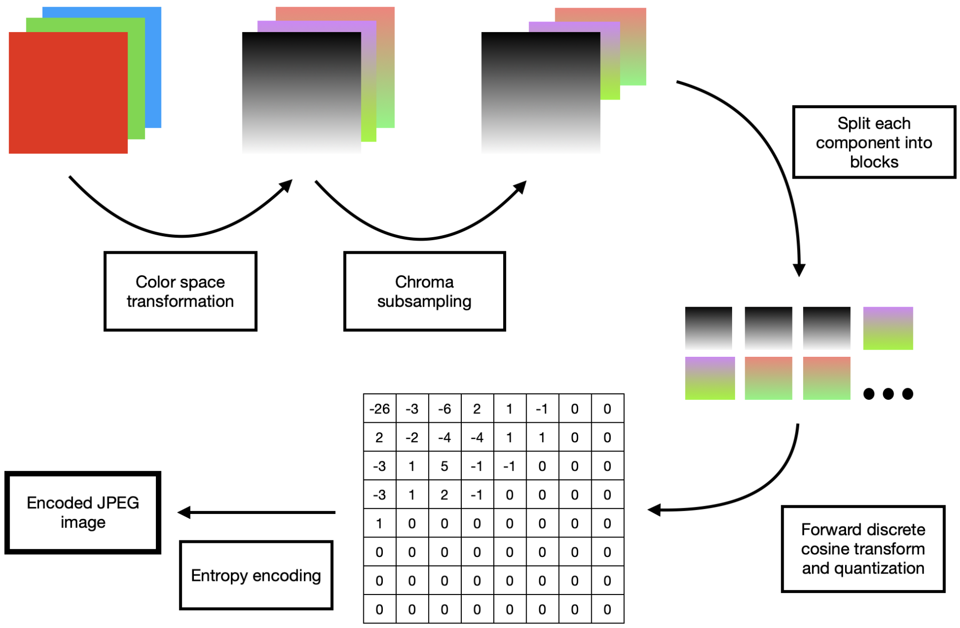


Figure 1: The main steps of the JPEG compression encoding process: 1) Color space transformation, 2) chroma subsampling, 3) forward discrete cosine transform, 4) quantization with a specific quantization matrix and 5) entropy encoding.

1.2 Chroma Subsampling

Chroma subsampling is a technique used in image and video encoding to reduce the amount of data needed to represent an image. By taking advantage of the human visual system (HVS), which is more sensitive to luminance (brightness) variances than it is to chrominance (color) variances, this technique samples the chrominance data at a lower resolution rate than the luma data. [1] [6] [10] [16] [11]

The chroma subsampling process mainly consists of two steps. In the first step, the image's color space is transformed from RBG to YCbCr, as illustrated in figure 2. RGB stands for the colors red, green and blue and it is a widely used model for producing a wide range of colors, mainly for color displays. On the other hand, YCbCr is a color space that separates the chrominance information (Cb and Cr components) from the luminance information (Y component) and it is often used for image compression. In the next step, the Cb and Cr components are sampled at a lower resolution than the Y component.

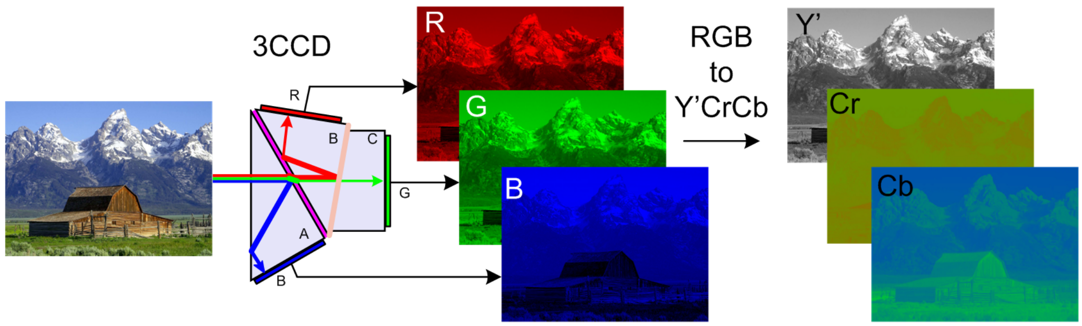


Figure 2: An image taken by a charge-coupled device in the RGB color space being transformed into YCbCr color space; Source: https://upload.wikimedia.org/wikipedia/commons/3/32/CCD.png - LionDoc, Public domain, via Wikimedia Commons - 12/11/2022

There are several subsampling schemes that one can use to compress an image and they are normally expressed as a three-part ratio J:a:b. Essentially, this notation is based on the concept of a reference block, which is a theoretical region comprising J image pixels in width and 2 image pixels in height. The letter "a" represents the number of chrominance pixels (Cb and Cr) sampled in the horizontal direction in the first row of "J" pixels in length. And the letter "b" represents the number of changes of chrominance pixels (Cb and Cr) sampled in the vertical direction between the first and second rows of "J" pixels in length. Some examples of subsampling schemes are illustrated in the 3.

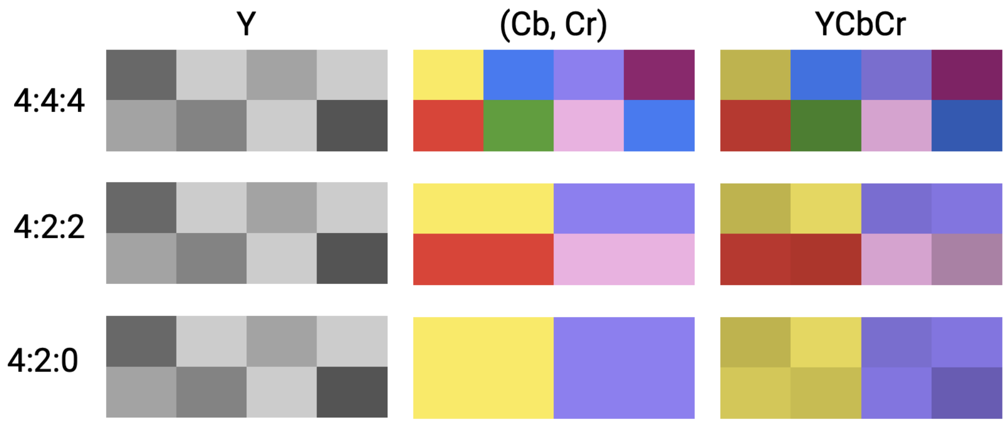


Figure 3: Example of chroma subsampling schemes: 4:4:4 : There is no chroma subsampling, since each of the three components Y, Cb and Cr has the same sampling rate. 4:2:2 : Every 4 horizontal sampling references, there are 2 chrominance samples (Cr, Cb) and 2 chrominance sample changes between the first and second row of pixels. 4:2:0 : Every 4 horizontal sampling references, there are 2 chrominance samples (Cr, Cb) but 0 chrominance sample changes between the first and second row of pixels.

1.3 Content Adaptive Subsampling

In addition to subsampling the chroma component, luma subsampling can also be performed, but it is usually not done due to the higher perceived loss of quality compared to uniform chroma subsampling, which limits the full compressibility of the images. Since the JPEG compression format already offers other means to reduce the image size without subsampling the luma information, such as through DCT and quantization, there is not much literature available that focuses primarily on subsampling the luma component to increase the compressibility of images.

William Bishop and Alexander Wong developed an algorithm that adaptively subsamples the Y component based on certain image content characteristics. The algorithm described in [16] is designed to increase the compressibility of images and videos and it works by analyzing the perceptual sensitivity of the HVS to certain content characteristics, such as texture activity, edge density and brightness.

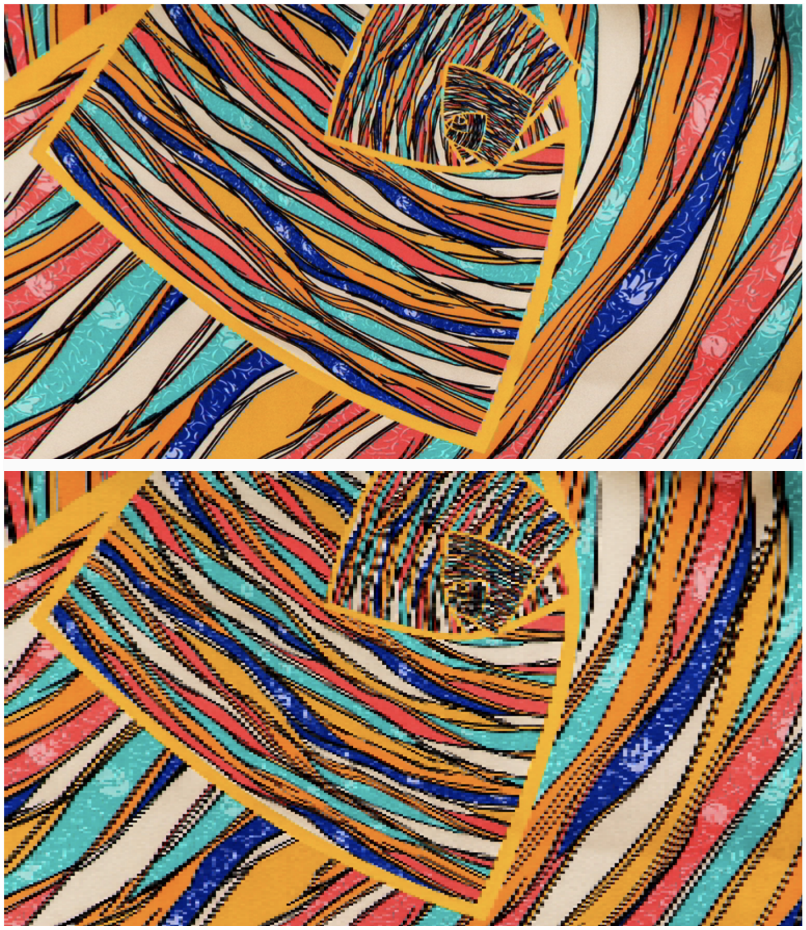


Figure 4: The first image was compressed using uniform chroma subsampling. The second image was compressed using uniform chroma subsampling and uniform luma subsampling. Both images were compressed using a 4:2:0 subsampling scheme. It is evident that uniformly subsampling the Y component can lead to poor image quality.

Wong and Bishop's objective was to develop a practical algorithm that is capable of subsampling the luminance information without significantly increasing the complexity of the code and that could also be used for video compression.

The proposed algorithm consists of several steps. The first step is to transform the color space of the image from RGB to YCbCr. As already mentioned, the color space transformation allows us to separate the luminance information from the chrominance information and evaluate and modify each component independently. Right after this step, the chroma components (Cb and Cr) are uniformly subsampled in accordance with the chosen subsampling scheme.

The second step is to divide the Y component into N blocks of size M by n based on the compression standard used. Once the Y component is divided into blocks, the content characteristics of each block can be analyzed. As mentioned above, three content characteristics are analyzed: edge density, average brightness, and texture activity.

1.3.1 Edge Density of the Block

As described in [16], edges play a crucial role in how the HVS interprets and perceives the environment and therefore, the HVS is very sensitive to edge degradation in images. To determine if the luma component of a block should be subsampled, edge characteristics should be considered.

To reduce visual degradation when sampling the luminance information, it should be only considered regions with a low concentration of edges. As described in [16], a threshold $T_{\text{edge}} = 0.1$ was chosen for each block. This means that blocks with an edge density rating $ER(x)$ below 0.1 are suitable for subsampling. This threshold was determined through subjective perceptual quality tests.

$$ER(x) < T_{\text{edge}}$$

To determine the edge density of a block, the entire image is processed using an edge detection algorithm such as the Canny edge detector. Once the edge pixels are mapped, it is possible to calculate the edge density by dividing the number of edge pixels by the total size of the block. For example, if a block with 100 pixels contains 9 edge pixels, the edge density is 0.09, making it a suitable candidate for subsampling based on the threshold of 0.1 mentioned previously.

The following is the equation to calculate the edge density of a block.

$$ER(x) = E_x/N_x$$

Where E_x is the number of edge pixels in block x and N_x is the total number of pixels.

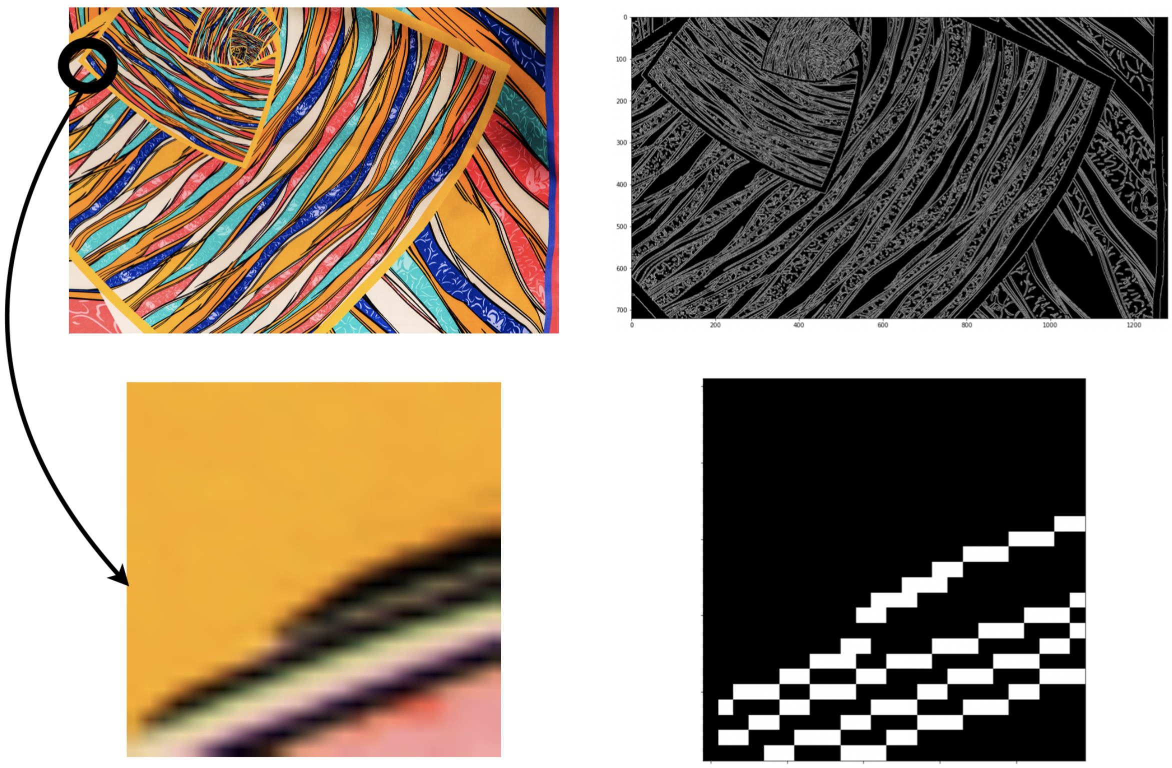


Figure 5: Example of the process to determine the edge density: In the first row, the canny edge detector is used in an abstract image, mapping all the edge pixels, represented as the color white. In the second row, it is possible to see the result of the canny edge detector for a block of size 25 x 25. For this block, there are 98 edge pixels, resulting in an edge density of 0.1568, above the threshold and therefore not a candidate for luma subsampling.

1.3.2 Texture Activity of the Block

Subsampled and reconstructed blocks with a low texture activity, such as smooth areas, can be decoded with less visual degradation compared to blocks with high texture activity. If an image has a lot of texture, visual degradation can be more noticeable when the image is subsampled, as the texture may become blurrier or more distorted.

According to [16], only blocks with low texture activity $TR(x)$ should be considered to be subsampled in order to minimize visual degradation. The spatial variance of pixel intensities in a block x is a useful metric for measuring the texture activity due to its low computational complexity.

$$TR(x) = s^2_{\text{pixel}}(x)$$

For testing purposes, the threshold of the variance is set at 0.001, for pixel values ranging between 0 and 1. If the variance of the pixel values within the block is less than 0.001, then the block should be considered to be subsampled.

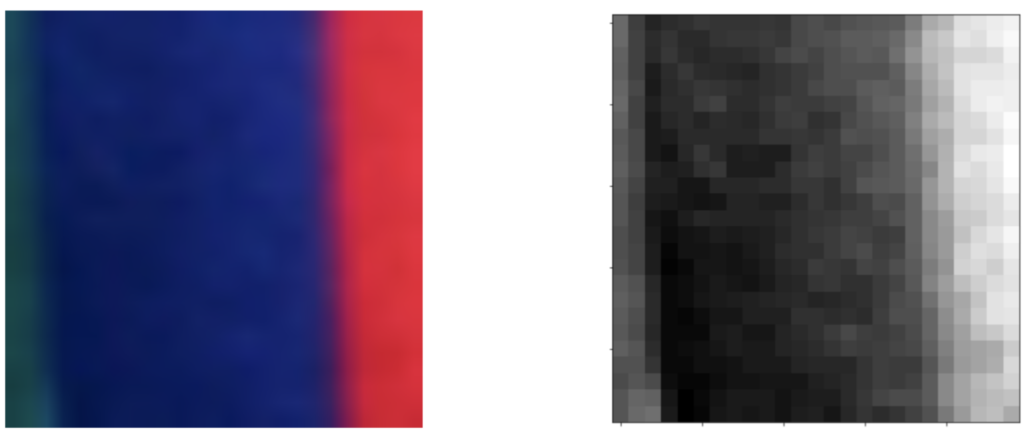


Figure 6: Example of texture activity: The Y component of an image block has its pixel values converted to the range between 0 (black) and 1 (white), and then the spatial variance of the pixel values is calculated. For this block of size 25 X 25, the variance is 0.006, meaning that this block should not be considered to be subsampled.

1.3.3 Mean Brightness of the Block

The HVS is less sensitive to details in dark areas, making them more able to conceal noise and degradation that may result from subsampling and therefore, blocks with a low brightness rating should be considered for subsampling [16].

The sampled mean can be used as a metric to calculate the overall brightness of a block $BR(x)$, since is a a simple and computationally efficient metric. Alexander Wong suggested a threshold of $T_{\text{brightness}} = 0.2$ for blocks with pixel values ranging from 0 to 1. Blocks with an overall brightness below 0.2 are suitable for subsampling. According to [16], "this threshold was determined based on several subjective perceptual quality tests, where perceptual degradation was noticeable if the threshold was set at a higher value".

$$BR(x) = \mu_{\text{pixel}}(x)$$

Where $\mu_{\text{pixel}}(x)$ is the sample mean of the pixel intensities in block x.

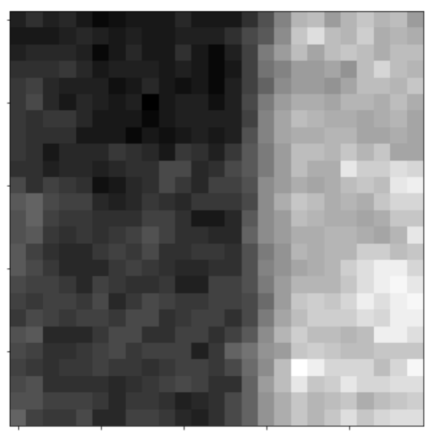


Figure 7: Example brightness: The Y component of an image block has its pixel values converted to the range between 0 and 1, and then the mean of the pixel intensities is calculated. For this block of size 25 x 25, the overall brightness is 0.17, which means that this block should be considered to be subsampled.

1.4 Luma Subsampling

For the Y component of a block to be considered a candidate to be subsampled, it has to satisfy the following criteria:

$$ER(x) < T_{\text{edge}} \text{ AND } (TR(x) < T_{\text{texture}} \text{ OR } BR(x) < T_{\text{brightness}})$$

All blocks that meet the criteria can have the Y component subsampled. Following the JPEG standards, which regulates that images are disassembled into blocks of 8 by 8 pixels in size [11] [13] [15] before they can be converted into a frequency domain representation using a discrete cosine transform (DCT), for testing purposes, I decided to use blocks of size 8 x 8.

Figure 8: Abstract image after the proposed content-adaptive subsampling algorithm. For demonstration purposes, the color of the blocks that had the Y component subsampled by the algorithm were changed to black. It is possible to see that the algorithm does not subsample the Y component of regions that don't satisfy the criteria, such as regions with a very high edge density.

1.5 Objective

The effectiveness of the algorithm proposed in [16] is evaluated solely using Peak signal-to-noise ratio (PSNR), which, as other objective measurement methods, has been shown to be an unreliable measure of perceived image quality, as mentioned in [14] and [7]. Additionally, the algorithm is tested on only two low resolution images, making it difficult to draw broader conclusions about its effectiveness. In this thesis, my objective is to subjectively evaluate the impact of the algorithm on perceived image quality using high definition images. For testing purposes, I used several subsampling schemes for different image categories. After analyzing the changes in perceived image quality, I also analyzed the increase in the compression factor for images compressed with the proposed algorithm and thus, being able to come to a deeper conclusion as to whether the algorithm is worth using.

My hypothesis is that, considering the content characteristics outlined in the paper to determine whether the Y component of a block should be subsampled or not, it is possible to apply the proposed algorithm and maintain a perceived image quality similar to that of images with uniform chroma subsampling for all subsampling schemes to be tested. I also expect that the compression factor increase directly depends on the image category.

2 Method

While objective measurement tests can measure certain technical aspects of an image, such as resolution or noise levels, they do not necessarily reflect exactly how the image looks to a human viewer. Subjective tests, on the other hand, involve humans being viewing and evaluating the images in order to assess their quality.

Wong and Bishop tested the proposed algorithm using Peak signal-to-noise ratio (PSNR) to measure the quality of the images. To calculate PSNR, the mean squared error (MSE) between the original and reconstructed signals is first calculated. The MSE is a measure of the difference between the two signals.

However, it is important to note that PSNR is not a perfect measure of image quality, as it does not take into account certain factors that are important to human perception such as color accuracy. As a result, subjective tests may be necessary to fully evaluate the quality of a reconstructed image.

To gain a more accurate understanding of the impact of the proposed algorithm on the perceived quality of the images, I conducted several subjective tests. These tests provide a more realistic assessment of image quality than relying solely on the Peak signal-to-noise ratio (PSNR) metric.

To simulate the content-adaptive luma subsampling algorithm, I wrote a program using Python that takes an image (original size or zoomed-in) as input, transforms the image's color space from RGB to YCbCr, subsamples the desired component (Cb, Cr or Y) with the desired subsampling scheme (4:4:2, 4:2:0 or 4:1:0), performs the discrete cosine transform and the quantization step and, in order to show the quality loss caused by the algorithm, the same steps mentioned above are performed in inverted order to decode the image and then the image is saved as a PNG file to ensure that there is no further data loss when saving the image. To measure the compression factor of the content-adaptive luma subsampling algorithm, I wrote a second program using Python that subsamples the desired component (Y, Cb or Cr) with the desired subsampling scheme and counts the total number of pixels for each component after chroma and luma subsampling and calculates the total size of the three components in bytes (each pixel is a 8-bit value) and the compression factor of the subsampled image.

2.1 Stimuli

To ensure a diverse and representative sample for the subjective tests, I selected eight high definition images (8-bit) from four different categories: Nature, Portrait, Sunset and Abstract. Using a variety of image categories allows for a more comprehensive evaluation of the proposed algorithm, as it helps to prevent any biases that may arise from using images with similar characteristics that are relevant to the algorithm. This allows for results that are more representative of the algorithm's performance in a wide range of real-world situations.

For the subjective tests, I included the original version of each image in its original size and a zoomed-in version of each image to enable a more detailed analysis of image quality when it is possible to see more details. I applied two different subsampling techniques to each version of image: uniform chroma subsampling and the content-adaptive subsampling algorithm. I also used three different subsampling schemes (4:2:2, 4:2:0, and 4:1:0) for both subsampling techniques, allowing for a comprehensive evaluation of image quality under a range of subsampling factors. The subsampling schemes 4:2:2 and 4:2:0 are the most used in the real world. This approach allows for a thorough analysis of the effects of different subsampling techniques and factors on image quality.

All images used in this experiment are license free and do not require attribution. It possible to find the images of the categories Nature, Abstract and Sunset on the following website: pixabay.com

The following are the images that I used for the subjective tests:

Figure 9: Image Abstract I

Figure 10: Image Abstract II

Figure 11: Image Sunset I

Figure 12: Image Sunset II

Figure 13: Image Nature I

Figure 14: Image Nature II

Figure 15: Image Portrait I

Figure 16: Image Portrait II

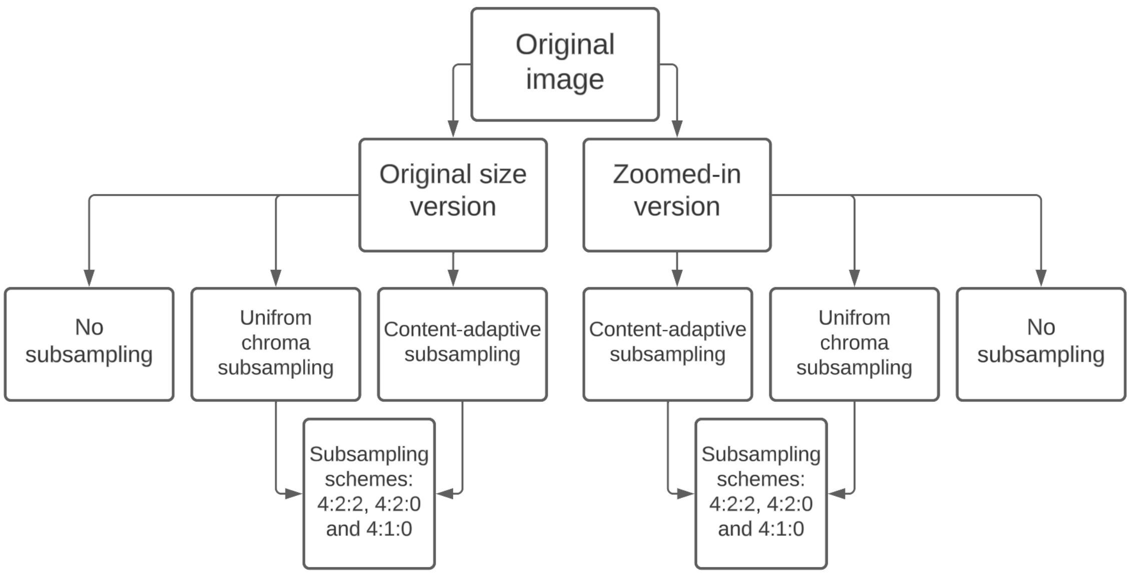


Figure 17: Stimuli: For the experiment, I used the original image as a full-size version and as a zoomed-in version and for each version there are 3 subversions: one without subsampling, one with uniform chroma subsampling and one with the proposed content-adaptive algorithm. For each subsampled subversion, I applied 3 different subsampling schemes.

2.2 Design and Procedure

To assess the perceived quality of each image, I used an Absolute Category Rating scale that utilizes a rating system ranging from 1 to 7, with additional areas at the ends of the scale (compared to the Mean opinion score "MOS") for subjects to use when they have particularly extreme perceptions. This helps to prevent saturation and reduces the tendency to avoid categories at the extremes of the scale, resulting in more accurate and comprehensive ratings, as described in [8].

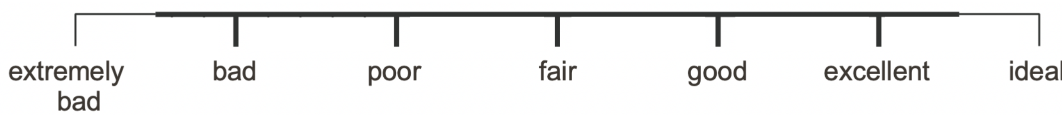


Figure 18: Absolute category rating scale used in the test subjects. "Extremely bad" is represented by the number 1, "bad" is represented by the number 2, "poor" is represented by the number 3, "fair" is represented by the number 4, "good" is represented by the number 5, "excellent" is represented by the number 6 and "ideal" is represented by the number 7.

The subjective tests followed the same rules:

1. Each test subject has a normal or corrected to normal visual ability.
2. The order of images was random and different for each test subject, thus avoiding biased results.
3. The subjective tests were performed with the same 14-inch monitor (3024 x 1964 Pixel) and using the same brightness configuration (full brightness) and thus avoiding very different results due to monitor influence.
4. The subjective tests were carried out in a calm room lit by daylight, trying to approach an environment with average daylight as described at ITU-R BT.500 from the BT Series [4].
5. The distance between each test subject and the monitor should

be between approximately 20 centimeters and approximately 60 centimeters, thus maintaining an approximate distance that is normally used with mobile devices and monitors.

6. Total of images: 8 images x 2 size versions (original and zoomed in) x 2 subsampling versions (uniform chroma subsampling and content-adaptive luma subsampling) x 3 subsampling schemes (4:2:2, 4:2:0 and 4:1:0) + 16 non-subsampled images (8 original size and 8 zoomed in) = 112 images

3 Results and Discussion

The table below shows the increase of the compression factor for each image category, separated by the subsampling scheme and the subsampling algorithm:

	Image category			
Type of Subsampling	Sunset	Abstract	Nature	Portrait
Chroma Sub. 4:2:2	1.5	1.5	1.5	1.5
Chroma Sub. 4:2:0	2	2	2	2
Chroma Sub. 4:1:0	2.4	2.4	2.4	2.4
Content Adap. Sub. 4:2:2	1.9	1.6	1.6	1.8
Content Adap. Sub. 4:2:0	3.5	2.2	2.2	2.95
Content Adap. Sub. 4:1:0	6	2.8	2.8	4.35

Table 1: Approximate values of the compression factors for each image category. Each value represents the arithmetic mean of all images in that category.

It is possible to see that, as expected, all images, regardless of the category, have the same compression factor for each subsampling scheme when using the uniform chroma subsampling algorithm. When analyzing the compression factors for the content-adaptive subsampling algorithm, it is possible to see some different results. Sunset and portrait category images have a significant increase in compression factor, with sunset category images being able to reach up to a compression factor of approximately 6 when using the 4:1:0 subsampling scheme and portrait category images being able to reach up to a compression factor of approximately 4.35.

Images from the portrait and abstract categories also demonstrate an increase in the compression factor in relation to the uniform subsampling algorithm. However, this increase is much smaller than that seen in the sunset and portrait categories. This is due to the fact that these categories have very different component characteristics. Images from the sunset category, for example, are much darker and have much less edges, which are important characteristics for the algorithm to decide whether it is possible to subsample the Y component.

Images from the abstract and nature category, on the other hand, have a very high density of edges, are brighter than images from the sunset category and have more texture activity, which leads to a smaller increase in the compression factor. In real life, the most used subsampling schemes are 4:2:2 and 4:2:0, and, as it is possible to notice from table 1, for these subsampling schemes, it is possible to achieve higher compression factors, which is one of the objectives of the proposed algorithm, but this increase depends highly on the image category.

To analyze the subjective test data, I decided to use box plots as well as boxenplots. A box plot is a graphical tool to represent the variation of the observed data of a variable by means of quartiles. Boxenplots show the distribution differently and are better for bigger datasets, as they can show more details about the distribution of data. Figure 19 and figure 20 show the overall result of the subjective tests for all image categories (original size and zoomed-in versions together).

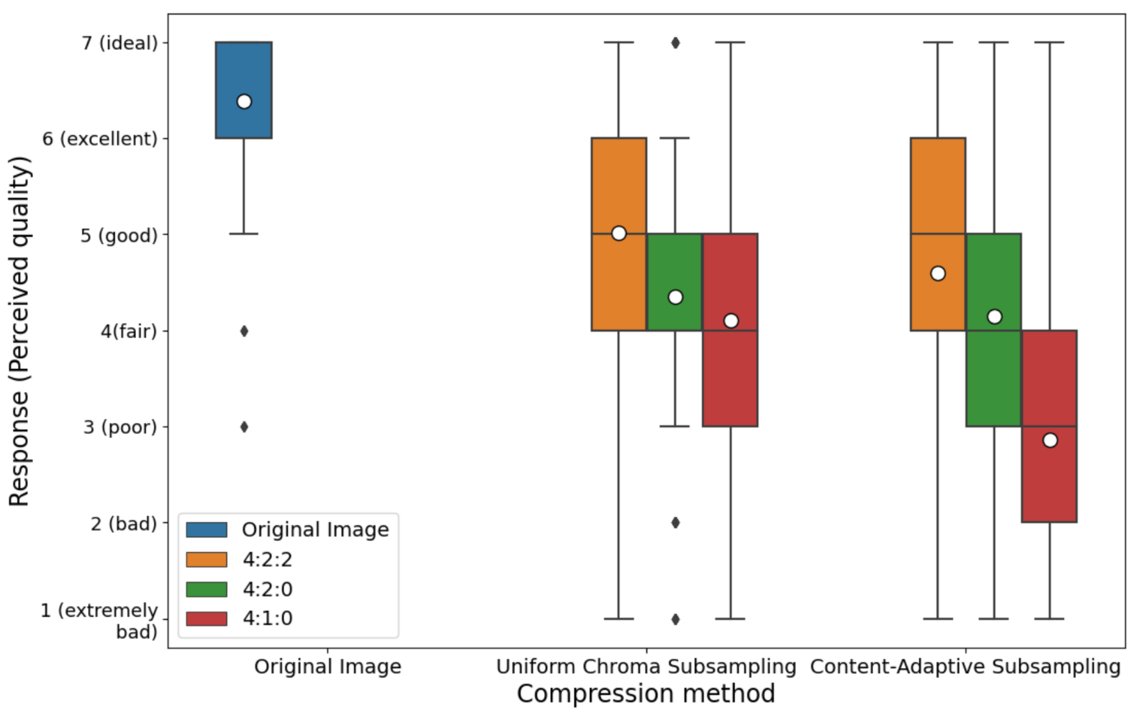


Figure 19: Overall result (original size and zoomed-in together). The X-axis represents the perceived quality. The Y-axis represents the subsampling algorithms used. The white dots represent the mean.

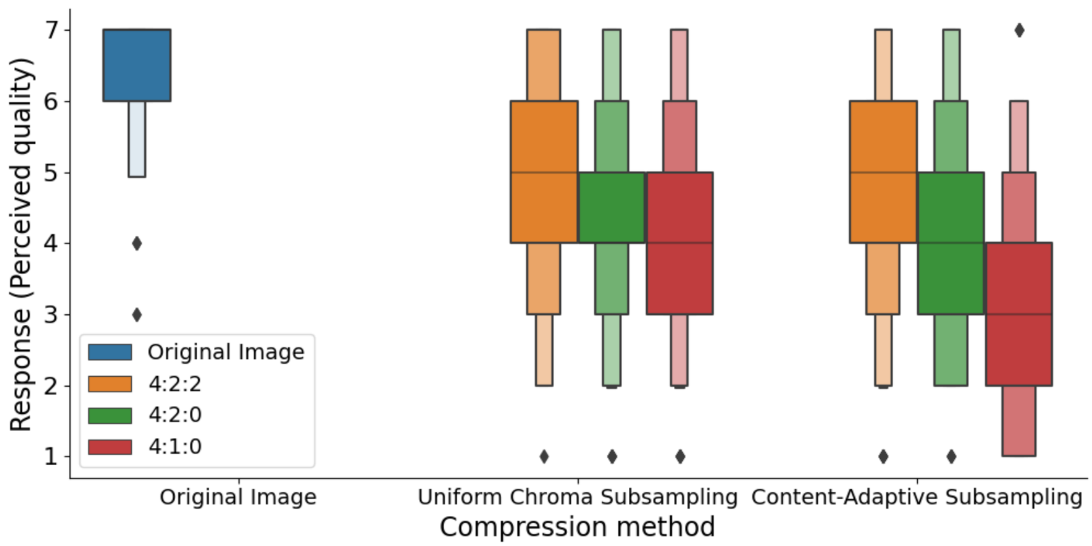


Figure 20: Overall result (original size and zoomed-in together). The X-axis represents the perceived quality. The Y-axis represents the subsampling algorithms used.

The original images, which are high resolution images, have very good visual quality as rated by the test subjects. Most people rated the original images as excellent or ideal. It is possible to notice that, as expected, the higher the uniform chroma subsampling ratio, the lower the perceived visual quality. For the 4:2:2 and 4:2:0 subsampling schemes, which are the most used in the real world, even if the averages of perceived image quality decrease, the averages remain above the value 4, which represents the category "fair", with even some people rating the perceived image quality as excellent or better(represented by the values 6 and 7).

When paying attention to the content-adaptive subsampling algorithm, it is possible to notice that for these same 4:2:2 and 4:2:0 subsampling schemes, the overall perceived image quality is very similar to the uniform chroma subsampling algorithm, especially for the scheme 4:2:2. For the 4:1:0 subsampling scheme, however, one can see that the perceived image quality for both algorithms drops even more in relation to the 4:2:0 subsampling scheme, but it is important to note that for the content-adaptive subsampling algorithm, the decrease is greater, with most people rating the perceived image quality between fair and bad. Figure 21 and figure 22 show the overall result for all image categories when in their original size only.

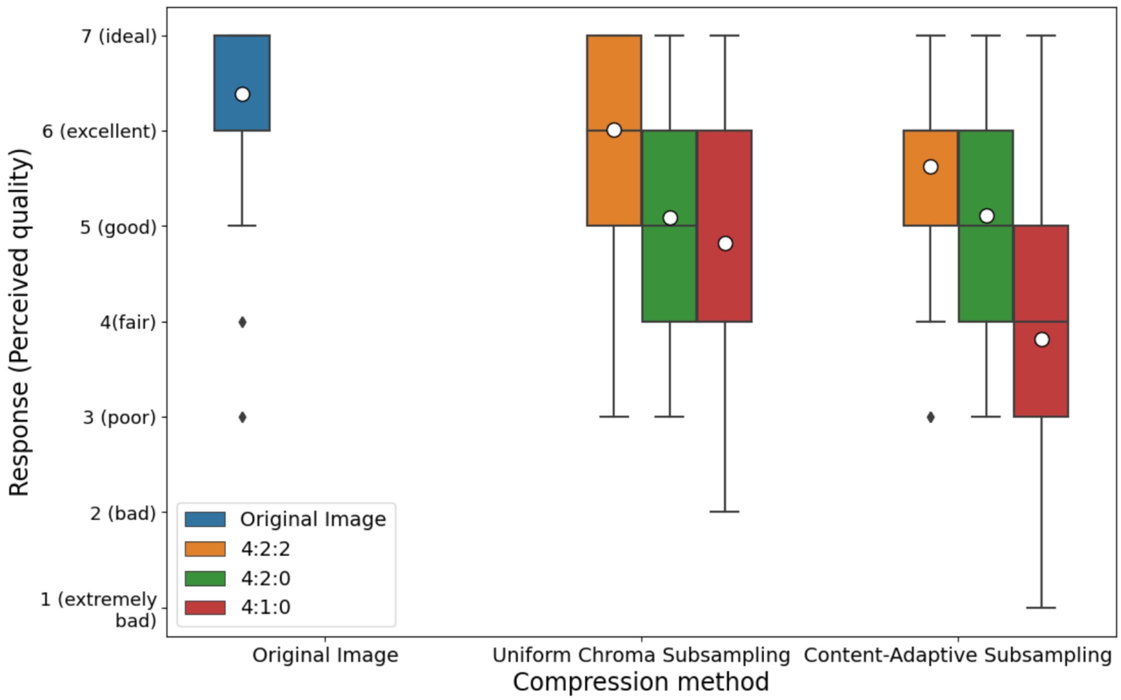


Figure 21: Overall result (original size only). The X-axis represents the perceived quality. The Y-axis represents the subsampling algorithms used. The white dots represent the mean.

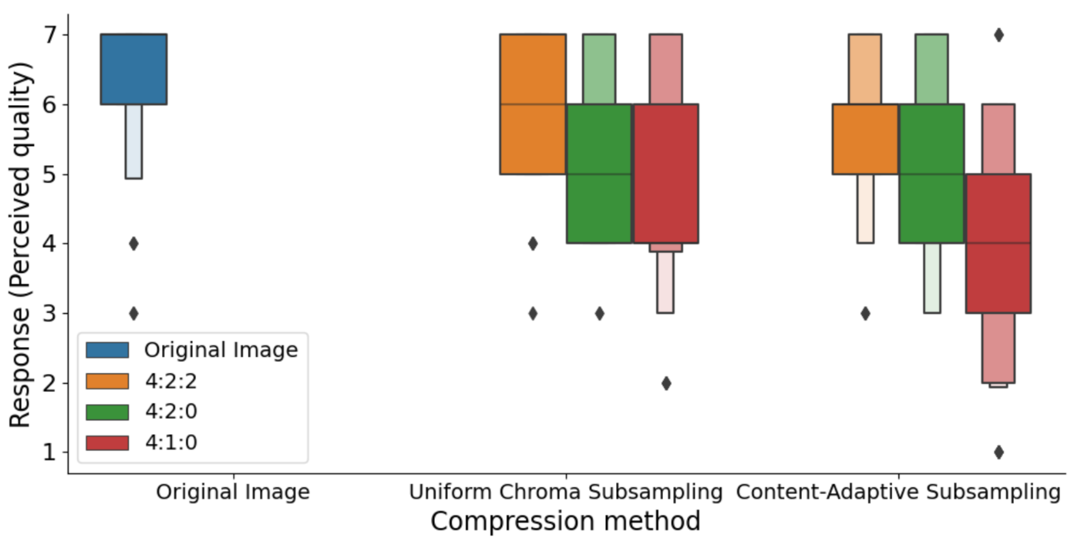


Figure 22: Overall result (original size only). The X-axis represents the perceived quality. The Y-axis represents the subsampling algorithms used.

The results illustrated here are very similar to the results shown in figure 19 and figure 20 for the overall results. Since we only analyze the images in their original size, we can expect better results than the results revealed in the previous charts, and that is exactly what we see here. Both algorithms result in a very similar visual quality, however, it is important to note, that for compressed images with the 4:1:0 subsampling scheme, even in their original size, when using the adaptive subsampling algorithm, the visual quality is worse than the visual quality resulted from the uniform chroma subsampling algorithm, with most test subjects rating these images as good, fair or poor and even some people classifying them as bad and extremely bad. Although the visual quality resulting from the adaptive subsampling algorithm is worse, the difference is not very significant and for the subsampling schemes 4:2:2 and 4:2:0, the perceived image quality is very similar to the uniform chroma subsampling algorithm.

Figure 23 and figure 24 show the overall result of the subjective tests for all image categories (zoomed-in only).

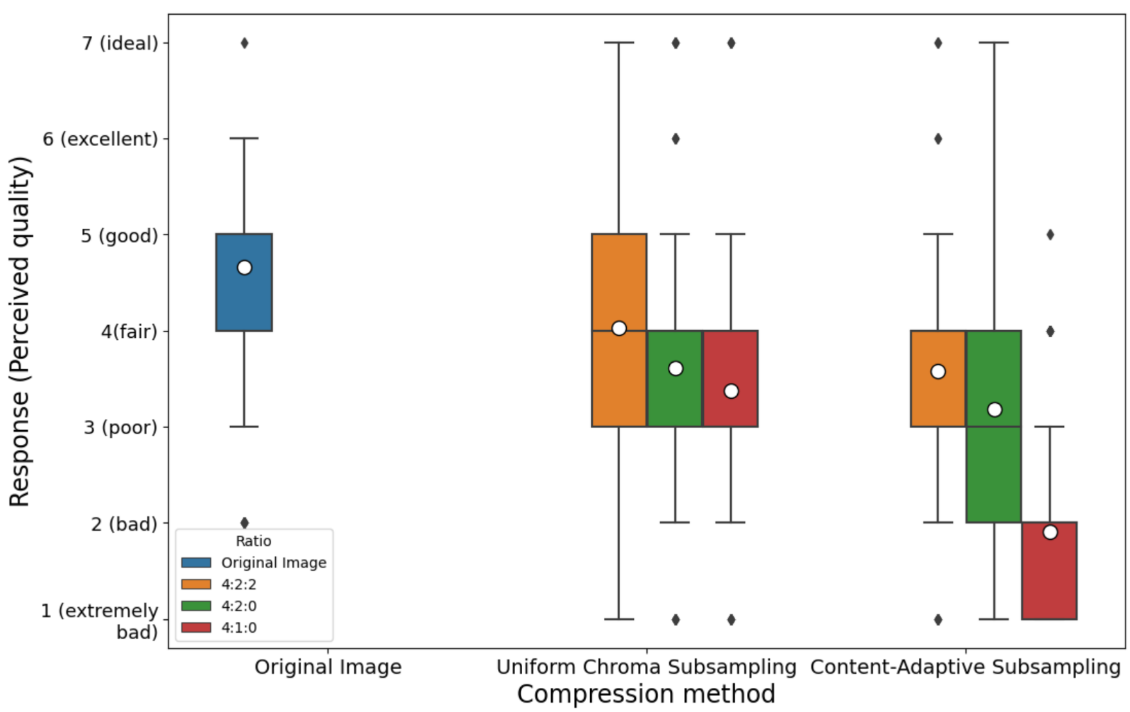


Figure 23: Overall result (zoomed-in only). The X-axis represents the perceived quality. The Y-axis represents the subsampling algorithms used. The white dots represent the mean.

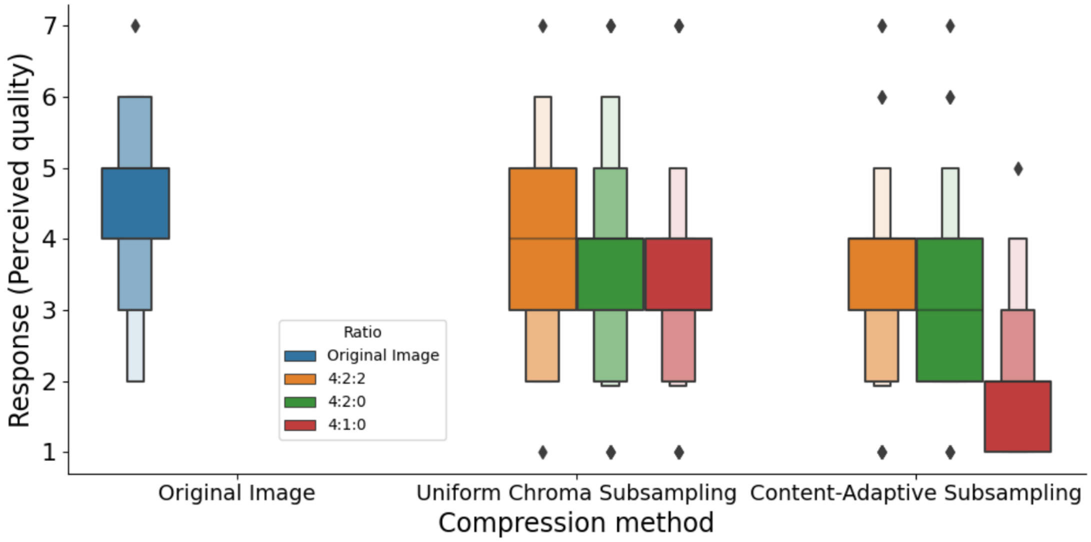


Figure 24: Overall result (zoomed-in only). The X-axis represents the perceived quality. The Y-axis represents the subsampling algorithms used.

These charts show that, when images are zoomed in, the perceived image quality when using the content-adaptive subsampling algorithm is worse than when using the uniform subsampling algorithm. With zoomed images, it is easier to notice the degradation of image quality, as it is easier to see the details of the images. We can see the visual quality deteriorate especially when using the subsampling scheme 4:1:0, where most test subjects rated the perceived quality as bad or extremely bad. This demonstrates that the algorithm proposed by Wong and Bishop works well for the subsampling schemes 4:2:2 and 4:2:0, especially when the images are in their original size, but when the images have the possibility to be zoomed in, then it is better to just use the 4:2:2 subsampling scheme and thus maintain an adequate image quality.

Figure 25 shows the result of the subjective tests for each image category in original size.

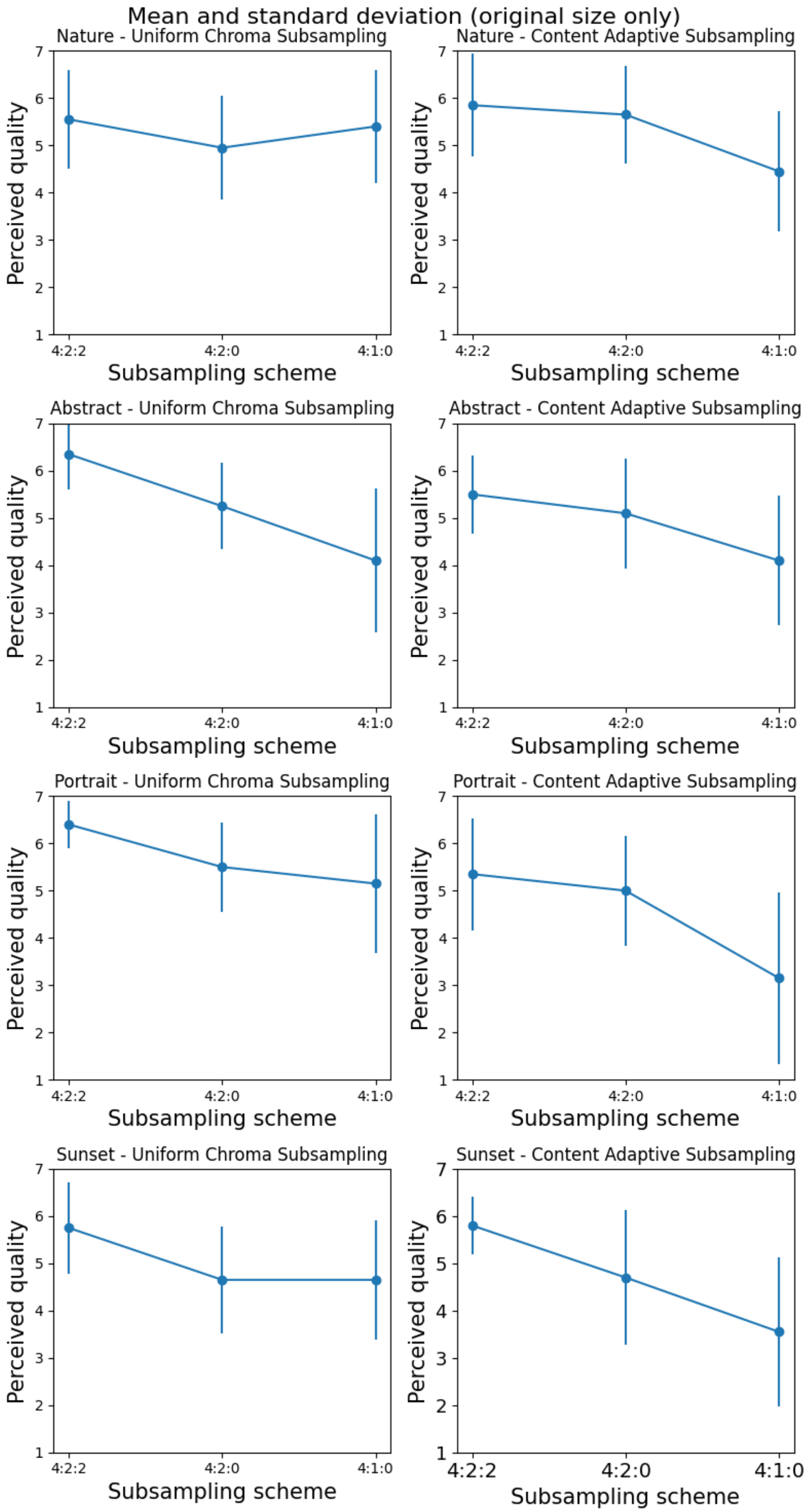


Figure 25: Results for each image category (original size). The X-axis represents the subsampling schemes. The Y-axis represents the perceived quality level. the blue dots are the means and the vertical lines are the standard deviation for that subsampling scheme.

As illustrated above, the image categories Sunset and Portrait, although not very significant, present a worse image quality than the other categories when using higher subsampling schemes ratios, especially 4:1:0. This correlates with the increase of the compression factor of images in these same categories. This means that, even though these categories demonstrate a significant increase in the compression factor, the visual quality deteriorates more than the other categories. But for the subsampling scheme 4:2:2, the perceived image quality of those images compressed by the proposed algorithm are very similar to those images compressed by the uniform chroma subsampling, which strengthens the argument presented above that the adaptive subsampling algorithm could be used to increase the compressibility of images while maintaining very similar image quality as when using the uniform chroma subsampling algorithm.

As David Salamon wrote in his book "Data Compression 3rd Edition": "The main aim of the field of data compression is, of course, to develop methods for better and better compression. However, one of the main dilemmas of the art of data compression is when to stop looking for better compression."

Although the proposed algorithm's runtime analysis and the code complexity analysis are beyond the scope of this thesis, both can only be expected to be higher when compared to the uniform chroma subsampling method. There is also the problem of real-world implementation of the algorithm, since it is not very feasible to change the standard encoding process.

I was able to confirm that the proposed content-adaptive luma subsampling algorithm could be used without significantly affecting the visual quality of images and that the compressibility of images also increases, leading to an increase in the compression factor, even though, depending on the image, the increase may be small.

In summary, the proposed content adaptive subsampling algorithm could be a suitable option for subsampling schemes with a low compression ratio, such as 4:2:2, which would produce a perceived image quality similar to uniform chroma subsampling. However, the compression factor increase may be minimal depending on the image. Although the increase in code complexity is not very significant, there is an increase and in an era where code efficiency is very important, it is difficult to imagine this algorithm being implemented in a permanent and global way. However, taking into account the theoretical aspects presented in this thesis, it is possible to state that there are other ways to increase image compression than that currently used in existing standard image compression formats.

4 Conclusion

This thesis subjectively evaluated the algorithm proposed by Wong and Bishop to adaptively subsample the Y component of images. The results show that, for high resolution images, the algorithm does not significantly degrade the image quality for small subsampling ratios and there is an increase in the compression factor. However, depending on the type of image, the difference in the compression factor may not be significant. Additional studies could explore different thresholds, as well as other content characteristics that could potentially allow for a higher compression factor without sacrificing perceived image quality or further increasing code complexity. Data compression is very important today and will continue to be for the foreseeable future.

5 References

[1] Glenn Chan. "Toward Better Chroma Subsampling: Recipient of the 2007 SMPTE Student Paper Award". In: *SMPTE Motion Imaging Journal* 117.4 (2008), pp. 39–45. DOI: 10.5594/J15100.

[2] Salomon David. *Data Compression. The Complete Reference*. 2004, pp. 1–11. ISBN: 0-387-40697-2.

[3] DAVID A. HUFFMAN. "A Method for the Construction of Minimum-Redundancy Codes". In: *ROCEEDINGS OF THE I.R.E.* 1952, pp. 667–673. DOI: 10.1109/JRPROC.1952.273898.

[4] *ITU-R BT.500-14, Methodologies for the subjective assessment of the quality of television images - page 5*. URL: https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-14-201910-I!!PDF-E.pdf.

[5] ISO ITU-T and IEC JTC. "Generic coding of moving pictures and associated audio information-Part 2: video". In: *ITU-T Recommendation H* 262 (1995), pp. 13818–2.

[6] Douglas A. Kerr. *Chrominance Subsampling in Digital Images*. 2012. ISBN: 978-3-662-56046-4.

[7] Pedram Mohammadi, Abbas Ebrahimi-Moghadam, and Shahram Shirani. "Subjective and Objective Quality Assessment of Image: A Survey". In: *Majlesi Journal of Electrical Engineering* 9 (June 2014), pp. 8–9.

[8] Sebastin Möller. *Quality Engineering Qualität kommunikationstechnischer Systeme*. 2014, pp. 43–50. ISBN: 978-3-662-56046-4.

[9] Mahdi O.A., Mohammed M.A., and Mohamed A.J. "Implementing a Novel Approach an Convert Audio Compression to Text Coding via Hybrid Technique". In: *International Journal of Computer Science Issues. 9*. 2012, pp. 53–55.

[10] P. Pokorny. "Lossy Compression in the Chroma Subsampling Process". In: *WSEAS TRANSACTIONS on COMPUTERS*. Vol. 15. 2016.

[11] Heinrich Hussmann Rainer Malaka Andreas Butz. *Medieninformatik, Eine Einführung*. 2009, pp. 89–106. ISBN: 978-3-8273-7353-3.

[12] *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025*. URL: https://www.statista.com/statistics/871513/worldwide-data-created/ (visited on 05/11/2022).

[13] G. Wallace. “The JPEG Still Picture Compression Standard”. In: *Communications of the ACM*. Vol. 34. 1991, pp. 30–44. DOI: https://doi.org/10.1145/103085.103089.

[14] Zhou Wang, Alan C. Bovik, and Ligang Lu. “Why is image quality assessment so difficult?” In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 4. 2002, pp. IV-3313-IV–3316. DOI: 10.1109/ICASSP.2002.5745362.

[15] Y. Wiseman. *The still image lossy compression standard – JPEG*. URL: http://u.cs.biu.ac.il/~wiseman/jpeg2.pdf (visited on 05/12/2022).

[16] Alexander Wong and William Bishop. “Practical Content-Adaptive Subsampling for Image and Video Compression”. In: *Eighth IEEE International Symposium on Multimedia (ISM’06)*. 2006, pp. 667–673. DOI: 10.1109/ISM.2006.128.

6 List of Figures