First Libyan International Conference on Engineering Sciences & Applications (FLICESA_LA)

13 – 15 March 2023, Tripoli – Libya

Image Compression using WebP and JPEG algorithms

Jazya Moftah Ahmad
Computer Science, Department, University Sirte, Sirte, Libya
E-mail jazamoftah@su.edu.ly

Abstract: Image compression is an important technique in digital image processing, as it allows for the reduction of the storage and transmission requirements of digital images. Two popular algorithms for image compression are the WebP algorithm and the JPEG algorithm. WebP is an image format that uses lossy compression techniques. In lossy compression, some data from the original image is lost in order to reduce the file size.

The JPEG algorithm, on the other hand, is a lossy compression technique that utilizes the discrete cosine transform (DCT) to transform an image into the frequency domain. By selectively discarding high frequency coefficients, the JPEG algorithm is able to achieve a higher compression ratio at the cost of some loss in image quality. The JPEG algorithm is well-suited for compressing images with smooth transitions, but it is less effective at compressing images with high levels of detail.

In conclusion, the WebP and JPEG algorithms are both widely used for image compression, but they have different strengths and limitations. The choice of which algorithm to use depends on the specific requirements of the application, such as the need for lossless compression or the importance of preserving image quality. In this paper, we compared between compression images using WebP and JPEG. The results showed that WebP is more efficient than JPEG compression.

I. Introduction

In computing, compression refers to the process of encoding data using fewer bits than the original representation. It is a way to reduce the amount of data that needs to be stored or transmitted, which can be useful in a variety of contexts, such as when data needs to be transferred over a network or when there is limited storage space available [1].

There are many different algorithms and techniques that can be used for data compression, and the choice of which one to use depends on the characteristics of the data and the requirements of the application. Some common types of compression include lossless compression, which preserves the original data exactly, and lossy compression, which sacrifices some of the original data in order to achieve a higher level of compression [2].

Compression can be applied to a wide range of data types, including text, images, audio, and video. It is often used in combination with other techniques, such as data deduplication and data encryption, to further reduce the size of the data or to protect it from unauthorized access [1,2,3].

Image compression refers to the process of encoding digital images using fewer bits than the original representation. It is a way to reduce the size of image files, which can be useful when storing or transmitting images, particularly if they are large or there is a limit on the available storage or bandwidth [4].

There are several different techniques that can be used for image compression, including lossless and lossy methods. Lossless image compression methods preserve the original image data exactly, while lossy methods sacrifice some of the original data in order to achieve a higher level of compression [5,6,7].

One common method of image compression is called JPEG (Joint Photographic Experts Group), which is a lossy method that is widely used for compressing In addition to reducing the size of image files, image compression can also be used to improve the performance of image-based applications, such as web browsers and image editing software. It can also be used to optimize images for display on different devices, such as smartphones, tablets, and desktop computers.

There are two main types of data compression: lossless and lossy. Lossless data compression preserves the original data exactly, while lossy data compression sacrifices some of the original data in order to achieve a higher level of compression [1,2,3].

Lossy data compression methods, on the other hand, are typically used for compressing audio, video, and image data, where some loss of quality is acceptable in exchange for a higher level of compression. One example of a lossy data compression method is the Joint Photographic Experts Group (JPEG) algorithm, which is commonly used for compressing photographs and other images.

Overall, data compression is a widely used technique that allows us to store and transmit data more efficiently, while still maintaining the integrity of the original data. Whether you are a data scientist, a network engineer, or a casual user, understanding the basics of data compression can be a valuable skill in today's data-driven world.

Image compression involves using fewer bits to represent digital images, which can decrease the size of the image files. This can be beneficial when storing photographs and other images. Other methods include GIF (Graphics Interchange Format) and PNG (Portable Network Graphics), which are both lossless methods commonly used for compressing graphics and images with a limited number of colors [8,9].

or sending the images, especially if the files are large or there are limits on storage or bandwidth.

There are several different techniques that can be used for image compression, including JPEG, GIF, and PNG. JPEG, which stands for Joint Photographic Experts Group, is a commonly used lossy method for compressing photographs and other images. GIF (Graphics Interchange Format) and PNG (Portable Network Graphics) are both lossless methods that are often used for compressing graphics and images with a limited number of colors. In addition to JPEG, WebP is an image format that uses both lossy and lossless compression techniques. In lossy compression, some data from the original image is lost in order to reduce the file size. This means that the image may not be an exact replica of the original, but the differences may not be noticeable to the human eye.

WebP uses a predictive coding scheme, which means that it predicts the values of pixels based on the values of neighboring pixels. This allows it to achieve a high degree of compression while still maintaining good image quality.

Image compression can not only decrease the size of image files, but it can also enhance the performance of image-based applications, such as web browsers and image editing software. It can also optimize the display of images on various devices, including smartphones, tablets, and desktop computers.

II. Literature review

In the literature, researchers have been working on techniques and algorithms that have been used to compress images [5,6,7]. In [5], the authors discuss the most popular techniques from the two types of techniques (lossy and lossless techniques). In addition, they present the advantages as well as the imitation of each technique. While the authors in [5] focus on image compression techniques in general, the authors in [6] consider only the lossless image compression techniques. In particular, they have conducted a comparison between the considered techniques, including the Huffman Code. In [10], the authors present a method for compressing images that combines Wavelet-based image coding with Huffman encoding to achieve additional compression. In [11], the authors aim to examine the Huffman coding technique, an additional method that can be used to eliminate unnecessary bits in data by analyzing various characteristics or specifications. In [12], the authors propose the use of simple coding techniques, such as Huffman, Discrete Wavelet Transform (DWT) coding, and a fractal algorithm, for image compression.

There are several works try to optimize existing algorithms. For example, in [8], the authors describe a method for detecting double JPEG compression using the same quantization matrix. Their algorithm is based on the observation that when a JPEG image is recompressed repeatedly with the same quantization matrix, the number of different JPEG coefficients (i.e., the quantized discrete cosine transform coefficients) between successive versions tends to monotonically decrease. This allows to detect when an image has undergone multiple rounds of JPEG compression with the same quantization matrix. In the same direction, the authors in [9], propose a simple yet reliable algorithm for detecting the presence of nonaligned double JPEG compression (NA-JPEG) in compressed images. Their aim is to enhance the image compression even more. Event approaches have been proposed to optimize the image compression of a particular type. For instance, in [13]a method is proposed for adjusting the order in which JPEG compression is applied to GPS images in order to improve the overall quality of the resulting images. Another example, the authors in [14], to improve image compression in deep neural network (DNN) applications, they have developed a framework called "DeepN-JPEG." This framework is specifically designed for DNN image compression.

III. Materials and Methods

3.1 .How the WebP algorithm Works

WebP is an image format developed by Google that uses lossy compression techniques to reduce the file size of images. It is designed to be faster and more efficient than other image formats, such as JPEG and PNG, while still maintaining the same or higher quality. WebP is supported by a wide range of modern web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge. It is especially useful for web developers because it can help reduce the loading times of websites and improve the overall user experience.

WebP uses a predictive coding scheme, which means that it predicts the values of pixels based on the values of neighboring pixels. This allows it to achieve a high degree of compression while still maintaining good image quality.

Here's how the predictive coding scheme works:

- The image is divided into blocks of pixels.

- For each block, a predictor function is applied to predict the values of the pixels in the block based on the values of neighboring pixels.
- The difference between the predicted values and the actual values is calculated and encoded.
- The encoded differences, rather than the actual pixel values, are stored in the image file.
- Next. the algorithm applies a discrete cosine transform (DST) to each block. this decorrelates the data and makes it more compressible .

WebP also uses techniques such as discrete cosine transform (DCT) and chroma subsampling to further reduce the file size. DCT is a mathematical transformation that converts the pixel values from the spatial domain (the grid of pixels in the image) to the frequency domain. This makes it easier to identify and remove redundant information from the image. Chroma subsampling reduces the resolution of the color information in the image, which can also help to reduce the file size.

Overall, the combination of predictive coding, DCT, and chroma subsampling allows WebP to achieve a high level of compression while still maintaining good image quality.

3.2 . Implementation of Image Processing using WebP algorithm (Lossy mode)

In the following part of the code, we show which libraries are needed.

```
image = Image.open('Image/T15.jpg')
# Start timer
start = time.perf_counter()
image = image.convert('RGB')
image.save('Image/T15.jpg.webp', 'webp',
optimize=True, quality=50)
# End timer
end = time.perf_counter()
# Calculate elapsed time
elapsed_time = end - start
print (f"Elapsed time: {elapsed_time:.6f} seconds")
print('done')
```

Figure 2: The main algorithm of the implemented WebP Algorithm

The images are loaded from a given path into the memory then, the image data is encoded. Finally, the encoded image data is written into a file. In the following, we show the main function that is used to compress images.

3.3 . Implementation of Image Processing using JPEG

To compress an image using the JPEG algorithm in Python, you can use a publicly available library. In particular, we use the pillow library to compress images.

(check the follow-up Python source code).

```
from PIL import Image
# Open the image file
with Image. Open('image.jpg') as image:
   # Set the compression quality to 50%
image.save('compressed_image.jpg', 'JPEG',
quality=50)
```

Figure3: Compress Images using pillow Library.

This code opens the image file image.jpg, sets the compression quality to 50%, and saves the compressed image as compressed_image.jpg. The compression quality is a value between 0 and 100, with higher values resulting in better image quality but larger file sizes, and lower values resulting in lower image quality but smaller file sizes.

IV. Evaluation and Results

In this section, we describe how we evaluate our approach of image compression using WebP algorithm. In addition, we compare it against the JPEG.

4.1. Evaluate WebP algorithm

We have evaluated the WebP algorithm using a set of images contain of 15 ones. These images are from different sizes. In the following table, we show all relevant related details for images. In particular, we show the original size of each image, as well as the compressed one. In addition, we show the compressed ratio. Furthermore, we show also the time needed to compress the images.

Table 1: Results of WebP Compression

Image name	**Original Size (kb)**	**Compressed Size (kb)**	**Differences (%)**	**Time of compression (s)**
IMG1	1800	229	87.2	0.72
IMG2	3900	1400	64.1	2.44
IMG3	344	98	71.5	0.16
IMG4	129	46	64.3	0.15
IMG5	279	20	92.8	0.10
IMG6	1800	528	70.7	1.27
IMG7	2700	224	91.7	0.57
IMG8	1900	278	85.4	0.67
IMG9	756	164	78.3	0.73
IMG10	2000	189	90.6	0.51
IMG11	193	32	83.4	0.10
IMG12	1900	244	87.2	0.99
IMG13	923	89	90.4	0.26
IMG14	1600	150	90.6	0.40
IMG15	1200	543	54.8	0.67

As we can see, all the images have been compressed. The ratio of the compression is different from one image to the others, which is expected, as a result of the image properties. In particular, the range of ratio is between 92.8% to 54.8%. Regarding the time that is needed to compress these images are listed also in the table. As illustrated, the time ranges between 0.10 seconds to 2.44 seconds.

. 4.2.Evaluate JPEG Compression

Similar to WebP algorithm, we also evaluated JPEG using the same set of images. These images are from different sizes. In the following table, we show all relevant related details for images. In particular, we show the original size of each image, as well as the compressed one. In addition, we show the compressed ratio. Furthermore, we show also the time needed to compress the images.

Table 2: Results of JPEG Compression

Image name	Original Size (kb)	Compressed Size (kb)	Differences (%)	Time of compression (s)
IMG1	1800	413	77.1	0.21
IMG2	3900	1600	59.0	0.63
IMG3	344	100	71.0	0.05
IMG4	129	70	45.7	0.03
IMG5	279	34	87.8	0.03
IMG6	1800	743	58.7	0.42
IMG7	2700	311	88.5	0.19
IMG8	1900	390	79.5	0.18
IMG9	756	398	47.4	0.20
IMG10	2000	276	86.2	0.16
IMG11	193	46	76.2	0.02
IMG12	1900	460	75.8	0.25
IMG13	923	127	86.2	0.08
IMG14	1600	196	87.8	0.10
IMG15	1200	594	50.5	0.15

As we can see, all the images have been compressed. The ratio of the compression is different from one image to the others, which is expected, as a result of the image properties. In particular, the range of ratio is between 88.5% to 50.5%. Regarding the time that is using WebP against the JPEG. The same set of images that have been used in our previous evaluation. In the follow-up, we have two columns. Each column shows the compression ratios of each algorithm. As illustrated in the following table as well as the related Figure 5, it clearly obvious that WebP is significantly more efficient than JPEG.

Table 3: Comparison of WebP and JPEG

Image name	WebP Compression (%)	JPEG Compression (%)
IMG1	87.2	77.1
IMG2	64.1	59.0
IMG3	71.5	71.0
IMG4	64.3	45.7
IMG5	92.8	87.8
IMG6	70.7	58.7
IMG7	91.7	88.5
IMG8	85.4	79.5
IMG9	78.3	47.4
IMG10	90.6	86.2
IMG11	83.4	76.2
IMG12	87.2	75.8
IMG13	90.4	86.2
IMG14	90.6	87.8
IMG15	54.8	50.5

4.3. Evaluate WebP algorithm Compared to JPEG

In this section, we show the result of our comparison. In particular, we compared the compression needed to compress these images are listed also in the table.

However, considering the time that is needed to compress images of each algorithm are listed in Table 1 and 2. In addition, we have illustrated the compared times in Figure 5, where we

show that the time is needed to compress images using JPEG is less than the time of the compressing images using WebP. Thus, JPEG needs less time to compress images than Web.

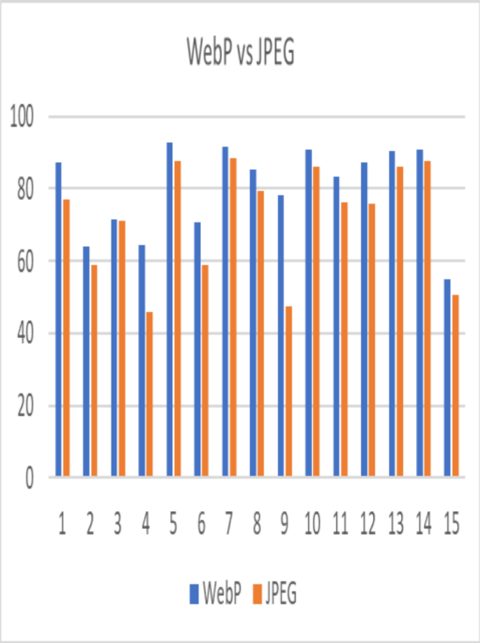

Figure 4: Compression ratio between WebP and JEPG

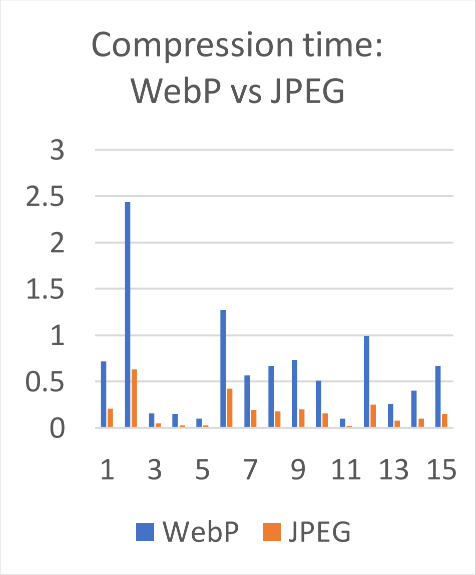

Figure 5: Compression time: WebP vs JPEG

V. Discussion

From the reported results above, we observe that WebP is significantly more efficient that JPEG. However, it looks that WebP need more time to compress images that JPEG. However, the range of the difference is between 0.1 to 2 second.

VI. Conclusion

Image compression is a crucial aspect of digital image processing as it enables the reduction of the storage space and

Image compression is a crucial aspect of digital image processing as it enables the reduction of the storage space and transmission needs of digital images. The WebP and JPEG algorithms are two well-known techniques used for image compression.

WebP is an image format developed by Google that uses lossy compression techniques to reduce the file size of images. It is designed to be faster and more efficient than other image formats, such as JPEG and PNG, while still maintaining the same or higher quality. WebP is supported by a wide range of modern web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge. It is especially useful for web developers because it can help reduce the loading times of websites and improve the overall user experience.

algorithm is a lossy image compression technique that utilizes the discrete cosine transform (DCT) to convert an image into the frequency domain. By discarding certain high frequency coefficients, the JPEG algorithm can achieve a higher compression ratio while sacrificing some image quality. This algorithm works well for compressing images with smooth transitions, but it is less effective at compressing images with intricate details.

Overall, the WebP and JPEG algorithms are commonly utilized for image compression, but they have varying strengths and limitations. The selection of the appropriate algorithm depends on the specific needs of the application, including the necessity for lossless compression or the significance of maintaining image quality.

In this article, we compared the efficiency of different image compression techniques: WebP and JPEG (a lossy technique). We found that the lossy techniques WebP is more efficient in terms of file size than JPEG.

References

[1].Sayood, K. (2017). Introduction to data compression. Morgan Kaufmann.

[2].Lelewer, D. A., & Hirschberg, D. S. (1987). Data compression. ACM Computing Surveys (CSUR), 19(3), 261-296.

[3].Salomon, D. (2004). Data compression: the complete reference. Springer Science & Business Media.

[4].Dhawan, S. (2011). A review of image compression and comparison of its algorithms. International Journal of Electronics & Communication Technology, IJECT, 2(1), 22-26.

[5].Vijayvargiy, G., Silakari, S., & Pandey, R. (2013). A survey: various techniques of image compression. arXiv preprint arXiv:1311.6877.

[6].Rahman, M. A., & Hamada, M. (2019). Lossless image compression techniques: A state-of-the-art survey. Symmetry, 11(10), 1274.

[7].Hussain, A. J., Al-Fayadh, A., & Radi, N. (2018). Image compression techniques: A survey in lossless and lossy algorithms. Neurocomputing, 300, 44-69.

[8].Huang, F., Huang, J., & Shi, Y. Q. (2010). Detecting double JPEG compression with the same quantization matrix. IEEE Transactions on Information Forensics and Security, 5(4), 848-856.

[9].Bianchi, T., & Piva, A. (2011). Detection of nonaligned double JPEG compression based on integer periodicity maps. IEEE transactions on Information Forensics and Security, 7(2), 842-848.

[10].Srikanth, S., & Meher, S. (2013, April). Compression efficiency for combining different embedded image compression techniques with Huffman encoding. In 2013 International Conference on Communication and Signal Processing (pp. 816-820). IEEE.

[11].Patel, R., Kumar, V., Tyagi, V., & Asthana, V. (2016, March). A fast and improved Image Compression technique using Huffman coding. In 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 2283-2286). IEEE.

[12].Jasmi, R. P., Perumal, B., & Rajasekaran, M. P. (2015, January). Comparison of image compression techniques using huffman coding, DWT and fractal algorithm. In 2015 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5). IEEE.

[13]Wiseman, Y. (2015). Enhancement of JPEG compression for GPS images. International Journal of Multimedia and Ubiquitous Engineering, 10(7), 255-264.

[14]Liu, Z., Liu, T., Wen, W., Jiang, L., Xu, J., Wang, Y., & Quan, G. (2018, June). DeepN-JPEG: A deep neural network favorable JPEG-based image compression framework. In Proceedings of the 55th annual design automation conference (pp. 1-6).