Lossless Image Compression

Kanisha Shah, Stuti Patel, *Member, IEEE,* and Prof. Twinkle Bhavsar

***Abstract*—The purpose of this paper is to provide an overview of image compression in various image formats. We tested compression ratios for jpeg, png, tiff, and bmp file extensions. Using MATLAB we created a graphical user interface to demonstrate the transition of the image before compression and after compression.**

***Index Terms*—Image Compression, Compression Ratio, Image Loss LaTeX, paper, template.**

I. Introduction

MULTIMEDIA pictures have become an indispensable part of our daily life. An image contains a significant amount of information. Thus, occupying space. Despite the advancements in bandwidth and storage capacity, many applications would be prohibitively expensive if images were not compressed. In this paper we have discussed the same issue keeping in mind that we do not loose any required information. Hence, making our compression lossless

November 20, 2021

II. Image Compression

Image compression is accomplished by eliminating all redundant data. To represent an uncompressed image, a large amount of data is required.

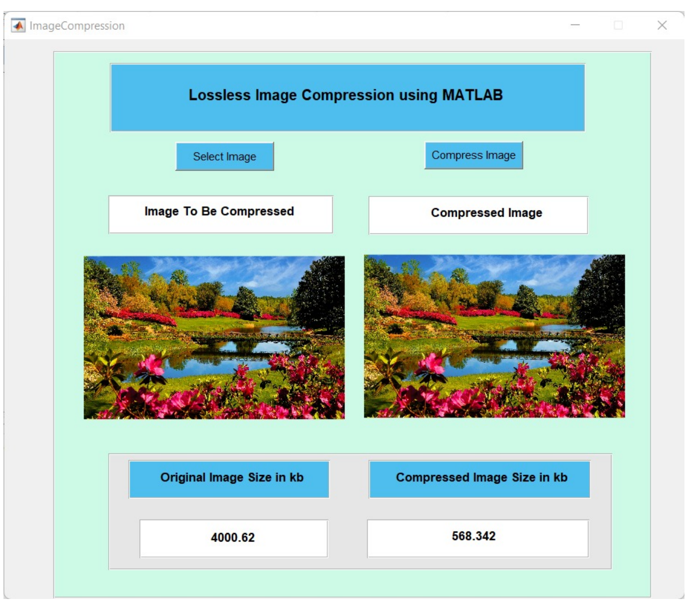

Fig. 1. Compression of JPG format

The above image required 3.9 MB on the system and after compression it just took 568 KB . The image was compressed by almost 7 times. Image compression is therefore important for effective image storage, retrieval, and transmission. Compression can be either lossy or lossless.

Fig. 2. Orignal Image

Lossy compression can benefit as it reduces the image size by several folds but information is lost too. In loseless compression

Fig. 3. Compressed Image

III. Compression Ratio

The compression ratio is equal to the original image size divided by the compressed image size. This ratio shows how much compression is done for a given image. Most algorithms have a set of compression ratios that they can achieve across a wide range of images. As a result, it's usually more practical to look at a new compression ratio for a certain method.

The compression ratio has a significant impact on the image quality. Generally speaking, the higher the compression ratio, the lower the image quality. When compressing photos, the trade-off between compression ratio and picture quality is critical to consider.

IV. Image Formats

A. PNG

As shown in Fig 4, our original image (in PNG format) was of the size 3595 Kb after compression we found it to be 233 Kb. Thus, compressing it by 15 times and retaining the quality.

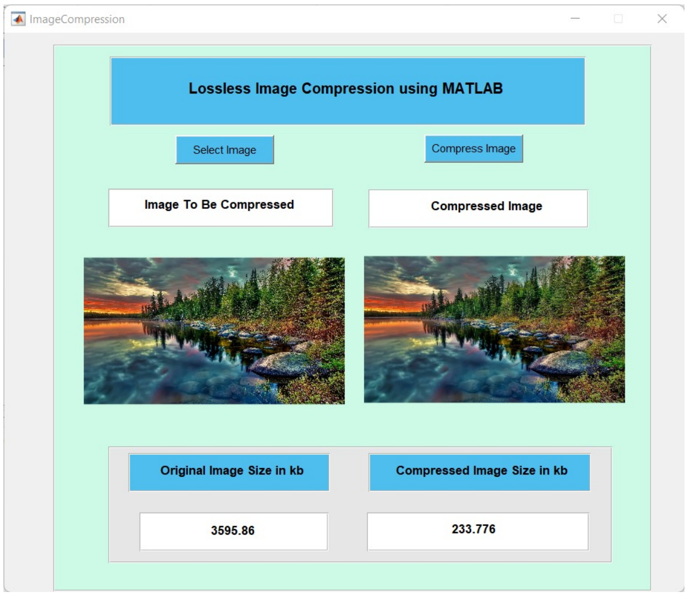

Fig. 4. Compression of PNG format

In Fig 5 we can see original image in .png extension. We can compare it with Fig 6 that is compressed image, we can clearly see the compression was truly lossless.

Fig. 5. Original Image in PNG format

B. TIFF

TIFF is Tagged Image File Format

Our original image (in TIFF format) was of the size 8960 Kb after compression we found it to be 476 Kb. Thus, compressing it by 18 times and retaining the quality.

In Fig. 7 we can see original image in .tiff extension. We can compare it with Fig. 8 that is compressed image, we can clearly see the compression was truly lossless.

Fig. 6. Compressed Image in PNG format

Fig. 7. Original Image in TIFF format

Fig. 8. Compressed Image in TIFF format

C. BMP

As we can see in Fig. 9, our original image (in BMP format) was of the size 1099 Kb after compression we found it to be 39 Kb. Thus, compressing it by 28 times and retaining the quality.

In Fig. 10 we can see original image in .tiff extension. We can compare it with Fig. 11 that is compressed image, we can clearly see the compression was truly lossless.

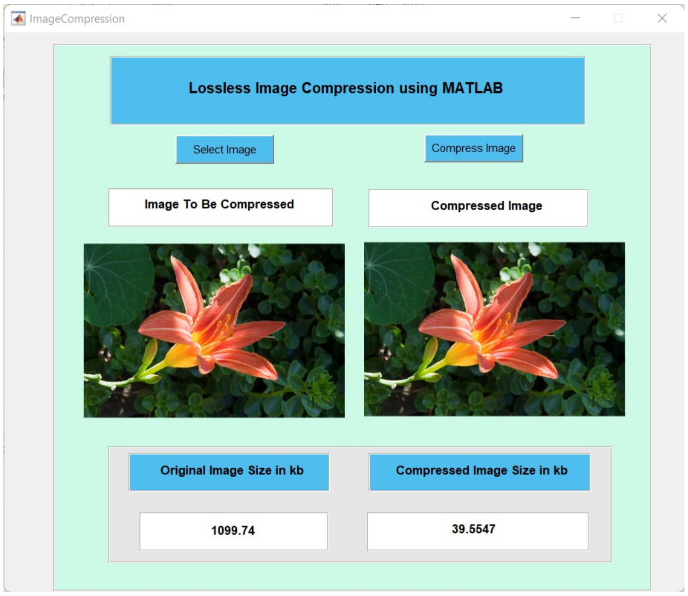

Fig. 9. Compression of BMP format

Fig. 10. Original Image in BMP format

V. Compression Technique

A. TIFF compression

TIFF (Tagged Interchange File Format) is a widely used and very versatile picture storage and sharing format that was created in 1995. It is widely used as a scanning output format and is used in many fax applications. The TIFF file format was created with three main aims in mind.

Fig. 11. Compressed Image in BMP format

- Extendable: This refers to the capacity to add new picture kinds while maintaining the functionality of existing ones.
- Portability: TIFF was created to be agnostic of the physical platform and operating system that it runs on. TIFF places low demands on the environment in which it operates. TIFF should (and does) work on a wide range of computer platforms, including PC, MAC, and UNIX.
- Reversibility: TIFF was created not only as an efficient means of sharing image data, but also as a native internal data format for image processing software.

TIFF supports a wide range of compression methods, including run length encoding, Huffman encoding, and LZW. TIFF is, without a doubt, one of the most adaptable compression formats. This algorithm's performance varies depending on the compression method employed. It's possible that it'll be either lossy or lossless. Another effect is that the length of time it takes to run varies depending on the compression strategy used.

TIFF has some drawbacks, such as the lack of support for vector graphics, text annotation, and other features (although such items could be easily constructed using TIFF extensions). The versatility of TIFF is maybe its biggest flaw. TIFF format, for example, can store data in both MSB ("Motorola") and LSB ("Intel") byte order, with a header item indicating which order is utilised. Keeping track of what's being used when can be fun, but it can also lead to code that's prone to errors.

The main advantage of TIFF is that it is a very versatile and platform-independent format that is supported by a wide range of image-processing software. It has a highly extensive space of information pieces for colorimetry calibration, gamut tables, and so on because it was designed by printer, scanner, and monitor producers. Remote sensing and multi spectral applications benefit greatly from this knowledge. The ability to breakdown an image by tiles rather than scan lines is another valuable feature of TIFF.

B. PNG compression

Compression of PNG images The PNG Development Group invented the PNG (Portable Network Graphic) picture format in 1995 as a replacement for GIF (the use of GIF was protested after the Unisys decision to start charging for use of the LZW compression algorithm). Instead, the LZ77 compression technique, which was devised in 1977 by Lemper and Ziv (without Welch) and improved in 1978, is used in the PNG (pronounced ”ping”) file format.

PNG is a free (for developers) open format with greater average compression than GIF and a number of unique features such as alpha transparency (so you may use the same image on many different-colored backgrounds). It also allows 24-bit images, which eliminates the need to index colours like in GIF. PNG is a lossless technique that follows many of the same guidelines as GIF. It takes $O(m^2 logm)$ time to run, where m is the number of colours in the image

VI. Source of Image Loss

JPEG is a lossy compression method. If you use this approach to compress an image, you can nearly guarantee that the decompressed version will not match the original source image. The algorithm's phases two and three are where information is lost. The discrete cosine transformation puts some inaccuracy into the image in phase two, but it is extremely minor. The inaccuracy is due to rounding and multiplication inaccuracies, and a considerable error is conceivable if the programmer chooses a DCT implementation that prioritises speed over quality. Any faults made during this phase have an equal chance of affecting any values in the image. It does not confine its mistake to a specific area of the image. Phase three, on the other hand, is intended to remove info that isn't essential to the image. In fact, this is where the majority of the loss in JPEG compression occurs. Each frequency value is quantized by dividing it by a constant and reducing the result. As a result, bigger constants generate more loss in the frequency matrix because the rounding error is greater. Because the higher constants are concentrated around the highest frequencies, and human vision is not extremely sensitive to those frequencies, the algorithm is designed in this manner. The quantization matrix is also customizable, allowing the user to control how much error is put into the compressed image. Obviously, the image size grows as the algorithm becomes less lossy. Users can commonly specify a value between 1 and 100 in applications that allow them to create JPEG images, with 100 being the least lossy. By most definitions, anything above 90 or 95 does not improve the image for the human eye, but it significantly increases the file size. Alternatively, excessively low numbers will produce exceedingly tiny files with a blocky appearance. In reality, several graphic artists generate stylistic effects in their work by using JPEG at very low quality settings.

VII. Conclusion

Format	Before Compression	After Compression	Compression Ratio
JPG	4000	568.34	7:1
PNG	3595.86	233.78	15:1
TIFF	8960	476	18:1
BMP	1099.74	39.55	27:1

All the value of image size are in kb, we can conclude that the highest compression took place when the extension was BMP followed by TIFF then PNG and then JPG.

Acknowledgment

We would like to thank our Prof. Twinkle Bhavsar, who gave us this amazing opportunity of writing a really good research paper on lossless image compression. We would like to thank our family who supported us while we were writing this.

References

[1] Lalit G. Patil, ”Lossless Image Compression using MATLAB”
[2] Matlab: https://matlab.mathworks.com/
[3] Images: https://favim.com/image/2857900/
[4] HD Images: https://www.pexels.com/collections/10-mb-apg68bg/