CS517

REPORT

LZW COMPRESSION

Presented By: Sahil

Entry No: 2021CSB1128

Introduction

In the field of digital image processing, the need for image compression arises due to the ever-increasing amount of digital image data being generated and shared on various platforms. High-resolution images occupy a significant amount of storage space and makes it cumbersome to transmit data. Image compression techniques are used to reduce the size of digital images while preserving the visual quality and details of the image. This reduction in size results in improved storage and transmission efficiency, making it easier to store, share, and process large amounts of digital image data. Various algorithms are used for image compression. This report mainly focuses on **LZW (Lempel-Ziv-Welch)** compression algorithm.

LZW (Lempel-Ziv-Welch)

LZW (Lempel-Ziv-Welch) is a lossless data compression algorithm that was developed by Abraham Lempel, Jacob Ziv, and Terry Welch in 1984. It is a dictionary-based compression algorithm that works by replacing sequences of characters with shorter codes, thereby reducing the overall size of the data.

The LZW algorithm is widely used in image file formats such as GIF and TIFF, as well as in other applications that require lossless data compression. Despite its effectiveness, the LZW algorithm has some limitations, such as its relatively high computational complexity, which can make it unsuitable for real-time compression applications.

INPUT AND OUTPUT

For Image compression using LZW, I have made the following functions, LZW_encoder() and LZW_decoder() which are used for encoding and decoding of image respectively. LZW is a dictionary based algorithm whose major principle is to match the previously occurred patterns. LZW_encoder() takes 3 arguments as input, filename, block size and the code size.

The default value of the block size is 8 which means the image is divided into 8 x 8 smaller matrices and these matrices are then encoded using this algorithm. We can set any value of the block size with -1 refers to the whole image. The code size here is the size of the

dictionary. Patterns whose index exceeds the dictionary size are not encoded. The encoded arrays are written in text file “lzw_encoded.txt”.

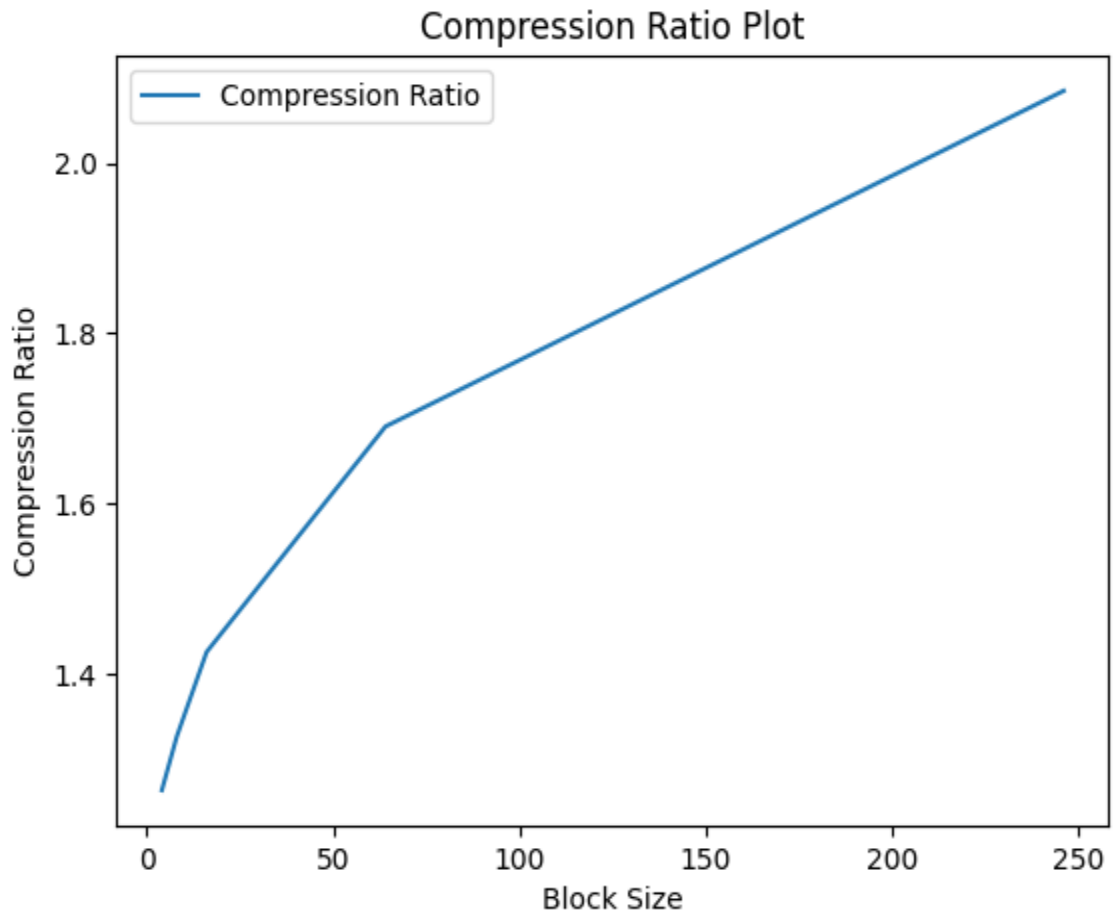
The decoder is used to decode the encoded text file. Since, LZW is a lossless algorithm, we get exactly the original image with rmse error 0. Several other parameters are also calculated like compression ratio, maximum value of code used in dictionary, entropy etc. The values we get helps to analyze the lzw algorithm in depth and shows why this algorithm is widely used for data compression.

ANALYSIS

Filename	Blocksize	CodeSize	RMSE	Compression Ratio	Max. Code Achieved	Entropy
book-cover	8 X 8	4096	0	1.504751	316	7.425825
checkerboard1024	8 X 8	4096	0	5.446808	264	1.0
Fig81a	8 X 8	4096	0	4.143717	280	1.661396
Fig81b	8 X 8	4096	0	1.882136	285	8.0
Fig81c	8 X 8	4096	0	1.771404	294	1.813469
fingerprint	8 X 8	4096	0	6.567906	304	6.567906
lena	8 X 8	4096	0	1.229702	317	7.383779
matches-aligned	8 X 8	4096	0	1.185636	317	7.346316
matches-random	8 X 8	4096	0	1.205092	317	7.425424
zoneplate	8 X 8	4096	0	1.507351	317	6.842690

We can clearly see that rmse value is 0 for each image. This shows that LZW is a lossless image/ data compression technique which is used very widely. Moreover, other parameters like compression ratio is always greater than 1 which indicates that image is compressed. For some images, compression ratio is very large which increases even further when block size is increased. This is because now more patterns can be matched and more encodings can be done.

- **GRAPH - COMPRESSION RATIO VS BLOCK SIZE**



OBSERVATIONS

- LZW compression works best on images with repetitive patterns, such as line drawings or charts. It may not be as effective on photographs or images with random patterns.
- LZW compression is a "dictionary-based" compression algorithm, which means that it creates a dictionary of frequently occurring patterns and replaces those patterns with a code that is shorter than the original data.
- LZW compression can achieve high compression ratios, especially for images with large areas of solid color.
- The size of the dictionary used in LZW compression can affect the compression ratio. A larger dictionary can lead to better compression but also requires more memory.
- LZW compression is widely used in file formats such as GIF and TIFF.

CONCLUSION

Overall, LZW compression is a useful technique for reducing the size of image files while maintaining their quality. It is particularly effective for images with repetitive patterns, and it can achieve high compression ratios. However, it may not be the best choice for all types of images, and the size of the dictionary used can affect the compression ratio. Nevertheless, LZW compression has found wide use in file formats such as GIF and TIFF, making it a valuable tool in image processing and storage.