Harshit Gupta(2021CSB1092)

LZW Compression

Introduction

LZW (Lempel-Ziv-Welch) is a lossless data compression algorithm that was first published in 1984 by Abraham Lempel, Jacob Ziv, and Terry Welch. It is a dictionary-based compression algorithm that works by replacing repeated sequences of symbols with a single code. It is a popular algorithm for compressing text,images and other types of data.

The LZW algorithm works by building a dictionary of symbol sequences that appear in the input data. Initially, the dictionary contains all possible symbols as individual entries. The algorithm then reads through the input data, looking for sequences that have

already been seen. When it finds a sequence that is not in the dictionary, it adds it to the dictionary and outputs the code for the previous sequence. The algorithm then restarts the process with the next symbol in the input data.
The biggest advantage of LZW compression over other compression is that it is a lossless compression i.e., the error between the original image and final decompressed image is zero

Some Common Terms

Compression Ratio

It represents the amount by which the original data is compressed, expressed as a ratio of the size of the compressed data to the size of the original data.
In the case of LZW compression, the compression ratio can range from 1.5:1 to 10:1 or more depending on the compression level. For example, a compression ratio of 10:1 means that the compressed data is one-tenth the size of the original data. The higher the compression ratio, the more the data is compressed, but also the more lossy the compression process becomes.

Entropy

The entropy of an image is a measure of the amount of information contained in the image. Entropy is a statistical measure that quantifies the randomness or unpredictability of a set of data. In the case of an image, the entropy can be calculated by treating the pixel values as a probability distribution and computing the entropy of that distribution.
We can compute the entropy using the formula:

$$H = -\Sigma(p * \log2(p))$$

Where p is the probability of occurrence of each pixel value in image

Terms used in Analysis

Total number of codes

This denotes the total number of values that we are sending in an compressed text file

Max value of codes used

This denotes the maximum value that we are sending in an compressed text file

Average length of encoded pixels

This denotes the average number of bits required to store code for one block

Analysis

Variation due to block size

As we increase the block size the compression ratio will increase, entropy will increase,average length of encoded pixels will increase,total number of codes decreases,max value of any code used increases

The various result of one of the sample images(book-cover.tif) when block size is changed is shown below:-

Block Size=4*4

```
Max value of any code used:  269
Total number of codes :  379513
Compression Ratio achieved :  1.2472405424847106
Average length of encoded pixels (average number of bits for each block) :  205.25310978907518
Entropy :  7.364731077574953
```

Block Size=8*8

```
Max value of any code used:  316
Total number of codes :  331717
Compression Ratio achieved :  1.426951286789643
Average length of encoded pixels (average number of bits for each block) :  717.6138453217956
Entropy :  7.742727326092429
```

Block Size=16*16

```
Max value of any code used:  495
Total number of codes :  288047
Compression Ratio achieved :  1.6432873801844838
Average length of encoded pixels (average number of bits for each block) :  2492.5646295294755
Entropy :  8.43110110400782
```

Block Size=32*32

```
Max value of any code used:  1093
Total number of codes :  252665
Compression Ratio achieved :  1.9615538361070983
Average length of encoded pixels (average number of bits for each block) :  8352.561983471074
Entropy :  9.35668273307772
```

Block Size=64*64

```
Max value of any code used:  3093
Total number of codes :  218989
Compression Ratio achieved :  2.263200434725032
Average length of encoded pixels (average number of bits for each block) :  28957.22314049587
Entropy :  10.512574866270167
```

Block Size=128*128

```
Max value of any code used:  8937
Total number of codes :  193167
Compression Ratio achieved :  3.053440805106462
Average length of encoded pixels (average number of bits for each block) :  85852.0
Entropy :  11.799507422553049
```

Block Size=whole image

```
Max value of any code used:  65535
Total number of codes :  142519
Compression Ratio achieved :  3.3212694447757842
Average length of encoded pixels (average number of bits for each block) :  2280304.0
Entropy :  14.21990376438759
```

Variation due to code size(maximum numbers bits used for coding)

For Small block size , larger code size will not help in increasing compression instead they will just reduce the compression. So, to utilize the functionality of variation of code size we should take a bigger block size. Here we are going to do analysis for block size of 128*128.

As the code size increases, max value of any code used will also increase,total number of codes decrease,compression ratio increases,average length of encoded pixels increases,Entropy increases

The various result of one of the sample images(book-cover.tif) when code size is changed are shown below:-

Code Size=9 bits

```
Max value of any code used:  511
Total number of codes :  410526
Compression Ratio achieved :  1.4367518744245187
Average length of encoded pixels (average number of bits for each block) :  102631.5
Entropy :  7.936752783792138
```

Code Size=10 bits

```
Max value of any code used:  1023
Total number of codes :  365503
Compression Ratio achieved :  1.6137323086267419
Average length of encoded pixels (average number of bits for each block) :  101528.61111111111
Entropy :  8.409130492603712
```

Code Size=11 bits

```
Max value of any code used:  2047
Total number of codes :  310619
Compression Ratio achieved :  1.898866456977841
Average length of encoded pixels (average number of bits for each block) :  94911.3611111111
Entropy :  9.008042713802528
```

Code Size=12 bits

```
Max value of any code used:  4095
Total number of codes :  214802
Compression Ratio achieved :  2.74589622070558
Average length of encoded pixels (average number of bits for each block) :  71600.66666666667
Entropy :  11.123159033626273
```

Code Size=13 bits

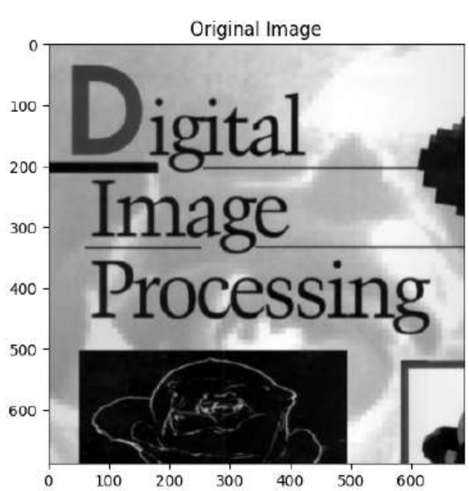
```
Max value of any code used:  8190
Total number of codes :  193670
Compression Ratio achieved :  3.0455104042959675
Average length of encoded pixels (average number of bits for each block) :  69936.38888888889
Entropy :  11.78542064028777
```

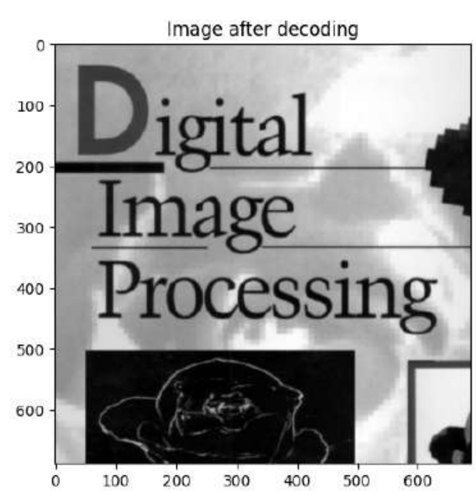
Code Size=14 bits

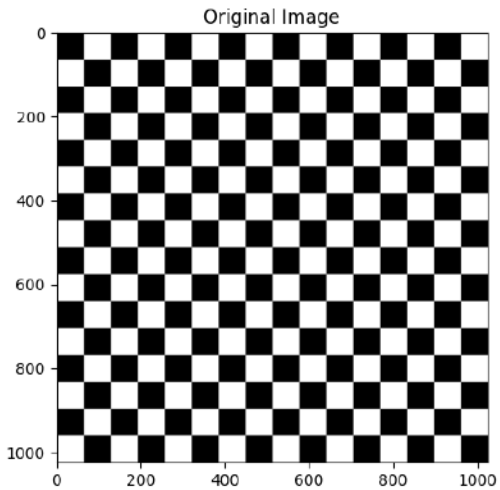
```
Max value of any code used:  8937
Total number of codes :  193167
Compression Ratio achieved :  3.053440805106462
Average length of encoded pixels (average number of bits for each block) :  75120.5
Entropy :  11.799507422553049
```

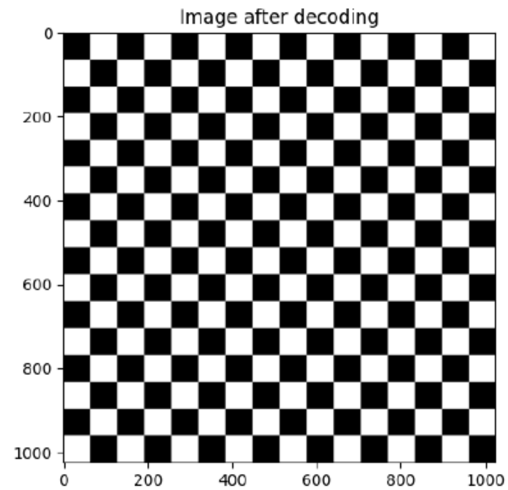
Result

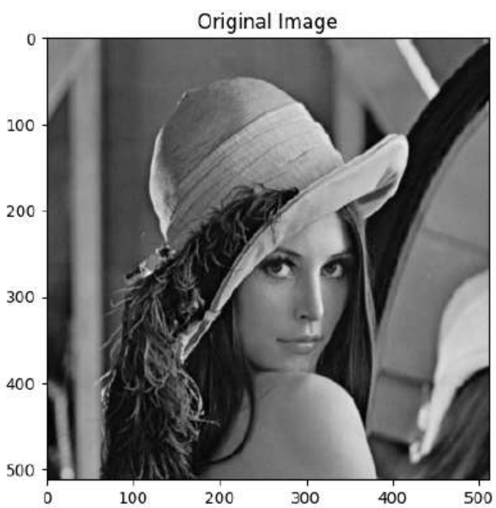
A few results of LZW compression are shown .We can see that the original and final decompressed images are exactly the same as this is a lossless compression and the RMSE between the original image and final decompressed image will be 0

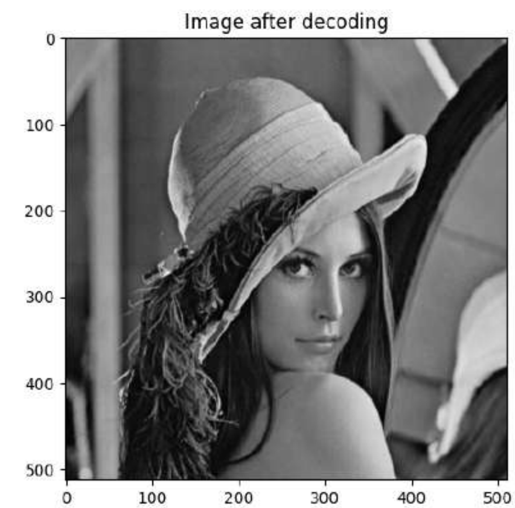












Observation

We have seen the variation of various parameters such as Max value of any code used, Total number of codes, Compression Ratio,Average length of encoded pixels.entropy when we vary our two input parameters i.e., block size and code size

Generally, we use block size of upto 64*64 and through data we have analysed that generally maximum value stays below 4095 upto block size of 64*64 . So , we can say that generally we need 12 bits of code to store the compressed image with block size upto 64*64

Conclusion

With the help of this project , we learned about the algorithms that were used when we perform LZW Compression. We also learned about the various of the output when we change various parameters like block size and code size.

We also verified that the RMSE between the original and the final image after decompression is 0 as this is a lossless compression

We also learned that LZW is not always the most efficient compression method for all types of data and may be less effective on already compressed data.