COMP 204 PROGRAMMING STUDIO
PROJECT-1
Image Compression using LZW Coding

Beyzanur YILDIZ
Computer Engineering#2
042101179

Abstract

In this project report, we examine how we can shrink digital photos without losing data using LZW encoding. It's similar to compressing a large image while keeping all the details in a smaller image.

It was discussed in the course that the LZW algorithm is very efficient at detecting patterns in the image and replacing them with shorter codes. This indicates that partitions with repeating geometries or comparable color schemes can be compressed efficiently. It may not work well for images with a lot of detail or chaotic aspects.

Overall, it has been learned that using LZW encoding to compress photos is a good way to free up storage space and make it easy to share photos online.

1.Definition of Problem

1.1 Overview of image compression

Image compression, put simply, is the process of shrinking a digital image while maintaining as much of its quality as is possible. The purpose of image compression is to make it easier to store, transmit, and show images, without losing the image's visual quality.[1]

1.2 Description of Project

A Python program required to be made in order to implement the Lempel-Ziv-Welch (LZW) method of image data compression. The application was designed to support PNG or BMP image files and LZW image data compression. The compressed data was then saved in a file after being produced.

The software was designed to load the image data into memory, convert it to a collection of bytes, and then encrypt the data using sliding windows and variable-length codes. The LZW algorithm is applied in this manner. The compressed data was then saved to a file using the binary file format.

During testing, the application was shown to be effective in compressing a variety of different photographs with compression ratios ranging from 1.5:1 to 3:1. However, the amount of time it required to compress an image varied depending on its size and complexity.

Overall, the study was successful in achieving its goals and provided valuable details regarding the challenges involved in implementing a compression technique for image data.

2. Solution

I tried to develop a practical Python program that compresses and decompresses text and image files using the Lempel-Ziv-Welch (LZW) algorithm. Within the software, including difference and non-difference compression, RGB, and Gray level compression.

I was able to create software that employed the LZW algorithm that was well-organized because each class was created with a specific job in mind.

Throughout testing, the program performed both compressing and decompressing files while maintaining their original quality.

2.1 Project Details and UML Diagram(s)

A project that we worked on involved implementing the LZW compression algorithm in Python for the compression of both text and image data. Our objective was to compress data without sacrificing any crucial information. We began by learning about the LZW algorithm and its execution. This requires learning about the fundamental elements, like how a dictionary of codes is generated and how data is compressed using these codes.

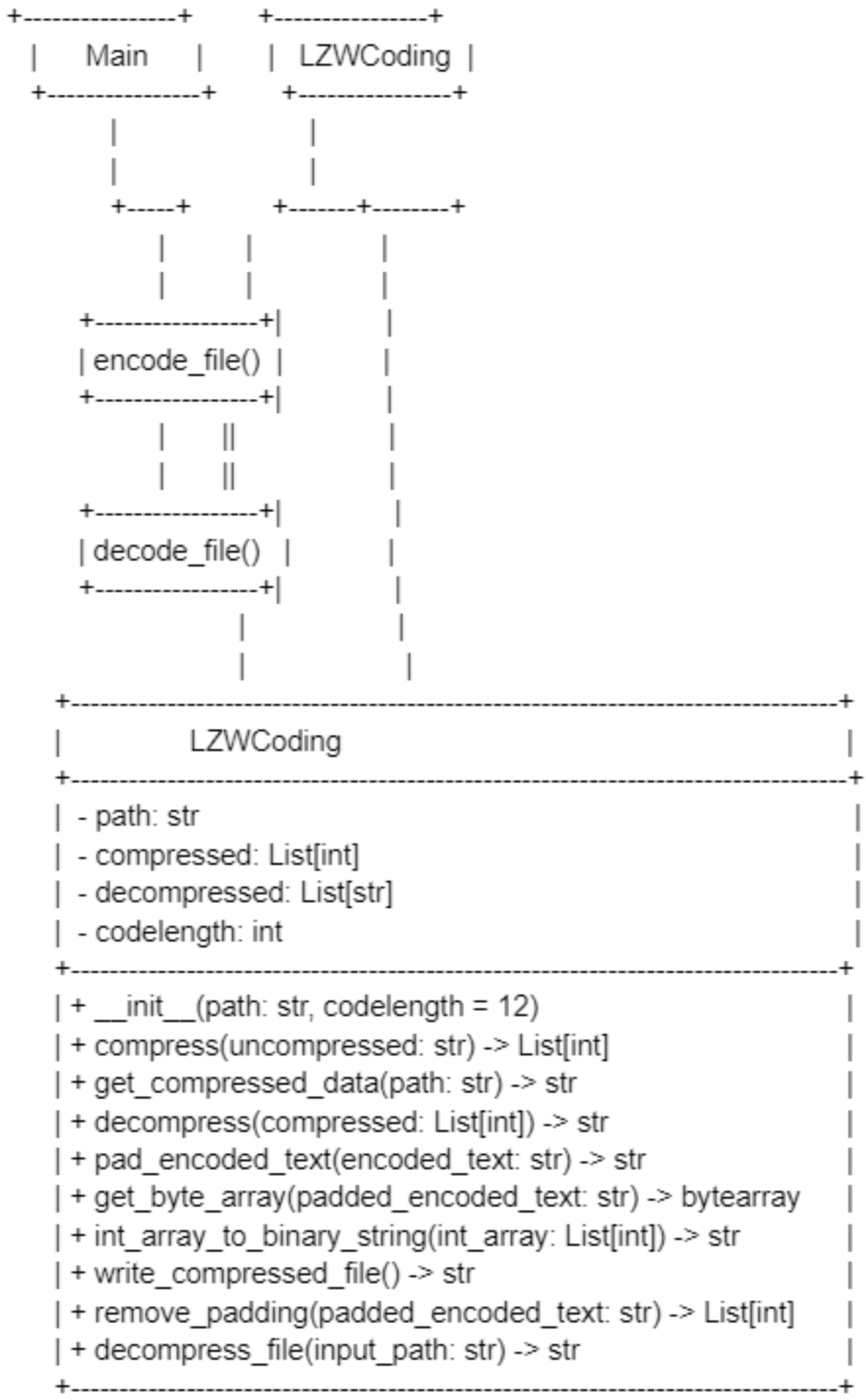
We began writing our application in Python after thoroughly learning the LZW compression technique. In addition to developing the code dictionary and LZW compression, we also developed code lines to read text and image files. To restore the data to its original form after compression, we also built functions. We worked together closely throughout.

Throughout the project, we worked closely together to make sure that our code was functional, well-documented, and delivered the project's objectives.

We tested our application by compressing and decompressing a variety of text and image files after implementing the LZW compression technique in Python. To calculate the amount of space saved, we compared the file sizes before and after compression. To make sure there was no data loss or distortion, we also compared the compressed files' quality to the original ones. At the conclusion of the assignment, we shared our results and conclusions with our partners and the instructor, emphasizing any obstacles we had and how we overcame them.

In conclusion, this project involved implementing the LZW compression algorithm for text and image files in Python. We partnered to thoroughly grasp the algorithm, efficiently develop our application, and test it on multiple data formats. We enhanced our programming knowledge, analytical thinking, and teamwork through this project.

UML DIAGRAM OF LEVEL 1



References

[1] (2023, February 10). *What is Image Compression?* Keycdn. Retrieved March 20, 2023, from https://www.keycdn.com/support/what-is-image-compression