Huffman Based LZW Lossless Image Compression Using Retinex Algorithm

[#1] R. Yeshwanth Raju, [#2] K. Nitin Reddy, [#3] M. Pavan Kumar, [#4] BK. Chinna Madhileti

[1,2,3] UG Student, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana

[4] Assistant Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana

Corresponding Author: R. Yeshwanth Raju, rajuyeshwanth11@gmail.com

Abstract—Image Data compression finds application in picture compression, where it minimizes the number of bits required to represent an image compared to its original format. In order to store and transmit data efficiently, picture compression aims to decrease irrelevant and redundant image data. Therefore, picture compression may speed up transmission and decrease network latency.In employing a lossless compression technique, data remains intact throughout the compression process without any loss. This study introduces a novel three-stage lossless compression approach called Huffman Inspired LZW Lossless Compression of Images utilising Retinex Algorithm. The first step is to compress the picture using Huffman coding. The second step is to compress the data using LZW and then decode it after joining all of the Huffman code words. Stage three involves applying the Retinex algorithm to the compressed picture in order to boost contrast and overall image quality. Using this suggested method, you can improve MATLAB's compression rate (CR), peak signal-to-noise ratio (PSNR), or mean square error (MSE).

Keywords— Lossless LZW Compression, Huffman Coding, Compression Ratio, Peak signal of Noise Ration, Retinex, Mean Square Error, LZW Coding and Decoding.

I. INTRODUCTION

By encoding the original picture with few bits, image compression applies data compression to images. The goal of picture compression is to make data storage and transmission more efficient by reducing irrelevant and redundant visual data. Compressing images entails retraining essential information while simultaneously decreasing the amount of the picture data. In mathematical terms, this entails converting a 2D array of pixels into a collection of data that is statically uncorrelated. The picture undergoes the alteration before it is stored or sent. In order to rebuild the picture or a close approximation of it, the image that was compressed is decompressed at a later time. Therefore, in order to reduce the amount of memory required to store a picture, image compression is used. It is impossible to transmit or store images that need an enormous amount of bits to be represented without lowering them in some way. Most photos have the trait of having correlated, and so redundant, information contained in nearby pixels. Finding a less associated representation of the picture is thus the primary objective. Redundancy & irrelevancy reduction are two essential components of compression. Eliminating duplicates in the signal source (picture or video) is the goal of redundancies reduction. Irrelevancy reduction removes information from the signal that is irrelevant to the receiver, in this case the eye of a human being.

II. RELATED WORK

A. Concise Overview of Common Methods and Standards for Image Compression:

With implications for many areas of image processing, this article provides a comprehensive overview of image compression methods and standards. In order to transmit and store images efficiently, image compression is crucial. This study introduces the Principal Component Analysis, a method for picture compression based on an analysis and evaluation of existing methods. There are a variety of methods for doing this compression; some use entropy coding to preserve all of the original image's information while others use lossy compression, which results in data loss. There are a number of other current standards for both still and moving picture compression outside of these methods, including as JPEG, JPEG2000, MPEG, and H.26x. In this research, I will compare and contrast some popular image compression techniques using a dataset of photos.

B. Compression through the use of Huffman coding:

As computer usage grows in new fields, more and more applications are relying on huge data systems, which in turn need the storage of massive data sets. Simultaneously, there is a tremendous increase in the transit of data via communication connections due to the growth of communication networks. One way to save money on storage and transmission is to compress data before storing or sending it. Decreases in the quantity of data that has to be communicated have the effect of making the communication channel more capable. In this paper, we provide a technique that efficiently decodes compressed data. An efficient Huffman decoding system is the focus of this work. The suggested model improves the decoding operation's speed. Xilinx tools are used for synthesis, placing and routing, floor planning, and simulation using Active HDL 5.1. The model is constructed using VHDL language.

C. Review of Images and Their compression Techniques:

Thanks to technical progress, which has no effect on picture management, advanced software tools that alter images, or the availability of many image operations, the accessibility of images in various applications is increased. There has been an outsized demand for storage capacity and communication bandwidth, even though both have been improved by technical advancements in transmission and storage. Because of this, picture compression has shown to be an effective method. Images and their compression methods are covered in this work. Our recommendations for the best picture compression algorithm are based on the review's overall criteria.

D. Lossless Compression of grayscale images utilizing the reduction of source symbols alongside Huffman coding:

Numerous apps have begun to make use of images, and their use has only grown. The ability to compress pictures before transferring them over a network is crucial for reducing file sizes and transmission times. The goal of this novel compressing method is to decrease the amount of source symbols while improving the compression proportion. Applying source symbols reductions and Huffman coding further reduces the source symbols to accomplish compression. The source symbols reduction method merges many source symbols into one new symbol, thereby reducing the amount of source symbols. So, it was also necessary to create fewer Huffman codes. With the decrease of Huffman code symbols, a greater compression ratio is achieved. Huffman coding and the suggested method were applied to standard pictures in the experiment. After looking at the data from the experiments, we can see that the newly suggested compression method attains a 10% higher compression ration than the standard Huffman coding.

E. The Huffman algorithm for double-compression of test data:

The quantity of test data is growing as a consequence of advancements in fabrication technologies and the complexity of designs. The main challenge in testing System-on-Chip (SoC) becomes the exponential growth of test size in relation to memory capacity. Multiple compression techniques have been proposed to reduce the volume of test data. One of these compression methods is code based approaches. When it comes to code-based compression, run-length coding is among the most used coding approaches. We may compress the test data and get a much higher compression ratio by using run length codes including Golomb codes, Frequency Directed Run Length Code (FDR code), Enhanced FDR, Adapted FDR, Shifted Alternate FDR, and OLEL coding techniques. A twofold compression strategy using the Huffman code is suggested for additional reduction of the input-test data. A comparison is made between the compression ratios achieved by various Run length coding and those obtained utilising the Double compression approach.

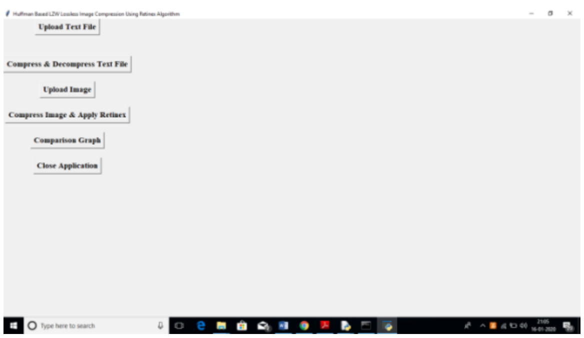
F. Image Compression with no Loss through combination:

The proliferation of digital photography and multimedia has resulted in an explosion in the amount of data needed to depict contemporary pictures. The storage space needed is substantial, and the time needed for transmission across computer networks is substantial, both of which are somewhat costly. All of these things point to the need of picture compression. To reduce the time and space needed for storing and transmitting digital images, image compression finds a solution by creating a smaller version of the original. The main point here is to decrease the size of a picture without losing any of the important information by removing redundant data from it. In this study, we focus on lossless picture compression. Our suggested method draws on a variety of established approaches. First, we subject the picture to the famous Lempel-Ziv-Welch (LZW) algorithm; this is the first step in our method. In the second stage, the results of the first stage are sent into the Bose, Chaudhuri, and Hocquenghem (BCH) algorithm for error detection and repair. The suggested method employs the BCH algorithms iteratively until "inflation" is identified in order to enhance the compression ratio. When tested against industry-standard compression methods, the experimental findings demonstrate that the suggested method can produce a superior compression ratio with no data loss.

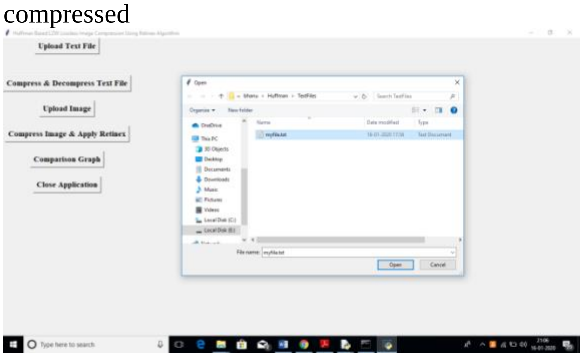
METHODOLOGY

1) **Upload Text**: We shall upload text using this module.
2) **Compress decompress Text:** This module will be used to compress and decompress text.
3) **Upload Image**: An picture will be uploaded using this module.
4) **Compress Image**: using this module, the image is compressed.
5) **Comparison graph**: We will compare the file sizes before and after compression using this module.
6) **Close application**: The application will be terminated when this module is used

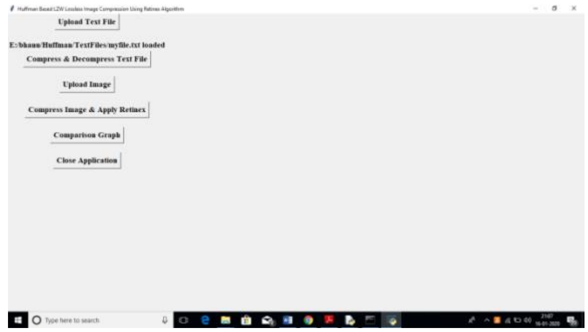
III. RESULT AND DISCUSSION



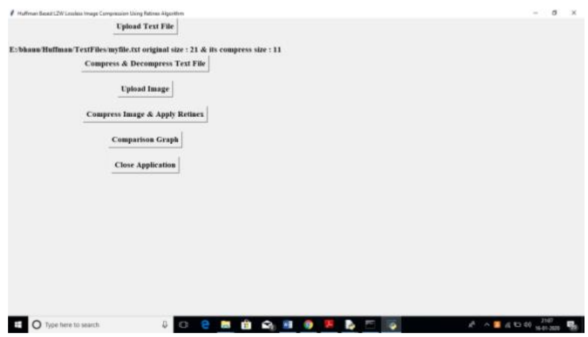
Click the "Upload Text File" button on the previous page to submit text files that you would want compressed



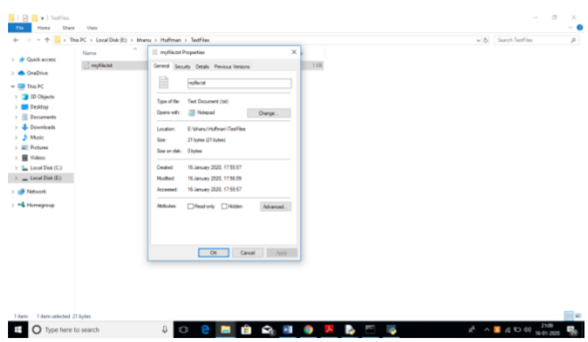
I am now uploading a file called "myfile.txt" to the screen up above; after it is uploaded, I will see the screen below.



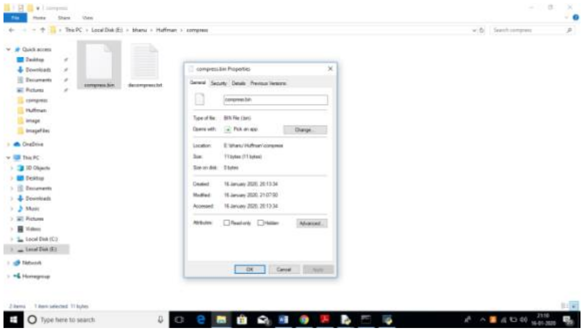
To learn how to compress and decompress a text file, click the "Compress & Decompress Text File" button.



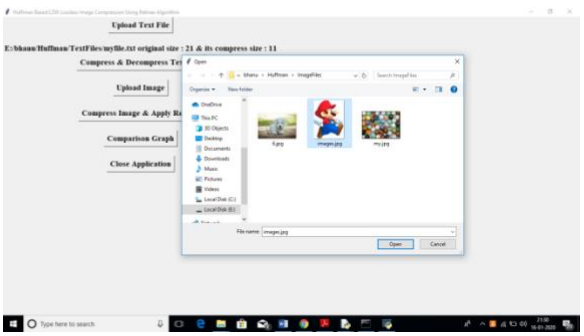
The notice is shown on the above screen as the compressed file size is 11 bytes, however the real file size is 21 bytes. The 'Text Files' folder now displays the original file size.



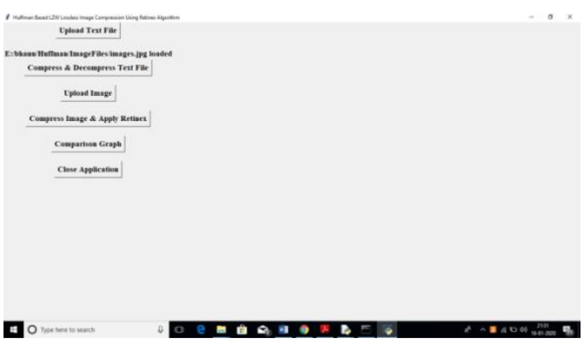
Now we can see the compressed file size within the 'compress' folder, as addition to the original file size in the directory, as seen in the previous screen.



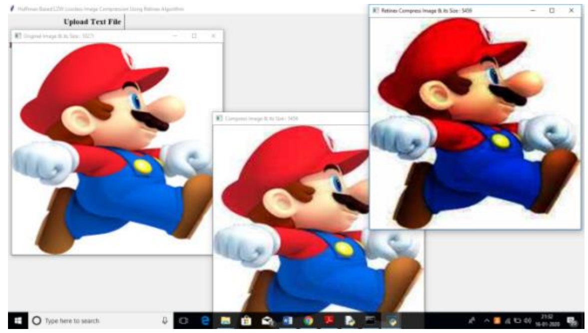
On the screen up there, compress. The decompress file includes the decompress data, while the bin file is 11 bytes in size. To upload a picture, do the same thing and click the "Upload Image" button.



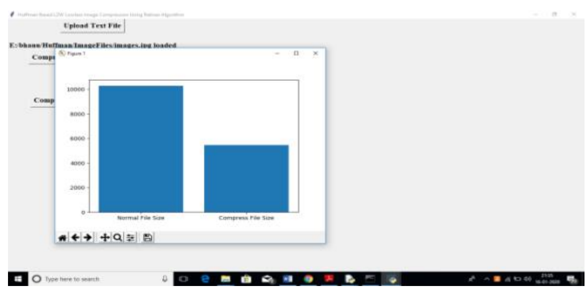
After you've uploaded the "images.jpg" file in the previous screen, you'll see the one below.



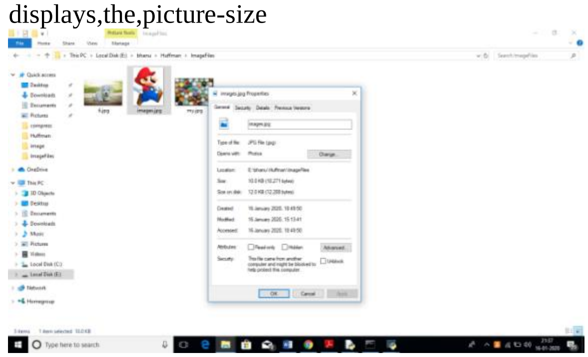
Then click on ‘Compress Image & Apply Retinex’ button to compress image



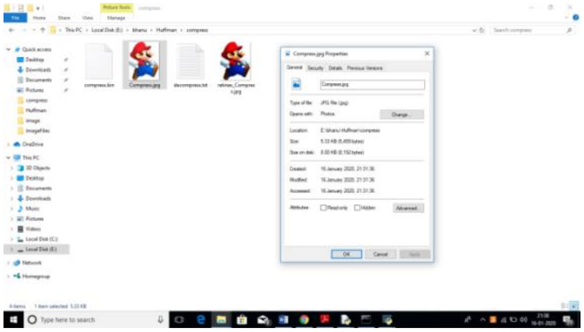
We can see the original image's name and size in the title bar of the first image; it's 10271 bytes in size. The second image is the compressed image; its size is 5459 bytes in size. The third image is the result of applying the Retinex algorithm to the compressed image; its size is also the same. After applying the algorithm, the third image looks slightly cleaner and brighter. The following graph will be appeared when you click the "Comparison Graph" button.



The y-axis in the following graph shows the size of the compressed picture, while the x-axis shows the regular image. We may lower the amount of the picture memory by applying compression, as seen plainly in the above graph. The directory now displays,the,picture-size



Pictured above are the originals. View the compressed file size in the compressed folder; the original jpg file is 10 KB in size.



Pictured above are the originals. View the compressed file size in the compressed folder; the original jpg file is 10 KB in size.

IV. CONCLUSION

With its wide range of uses, compression is an important and consequential subject. This thesis showcases the flawless image compressing method applied to various pictures. A number of methods have been tested for their compression levels, efficiency, and error proneness. The degree of compression accomplished is highly reliant on the source's qualities, in contrast to algorithm's efficiency and error susceptibility, which are somewhat independent of the trait of the source ensemble. Further compression is possible with additional data redundancy, it is determined. Since the Retinex Algorithm uses MSR to improve picture contrast, the quality of the reproduced image is same to that of the original.

VI. REFERENCES

[1] Vartika Singh "A Concise Overview of Common Methods and Standards for Image Compression" The 2013 Volume II edition of the International Journal of Technology and Research Advances.

[2] MamtaSharmap "Compression through the use of Huffman coding" The May 2010 issue of VOL.10 No.5 of the IJCSNS International Journal of Computer Science and Network Security.

[3] Sindhu M and Rajkamal R's paper titled "A Review of Image Compression Techniques" published in the International Journal of Recent Trends in Engineering, Volume 2, Issue 4, November 2009.

[4] C. Saravanan and R. Ponalagusamy's research on "Lossless Compression of Gray-scale Images via Reduction of Source Symbols Combined with Huffman Coding."

[5] MayurNandihalli, VishwanathBaligar "Lossless Compression of Gray Scale Images Utilizing Dynamic Arrays for Prediction, Focusing on Higher Bit Planes" published in the International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.

[6] R.S. Aarthi, D. Muralitharan, and P. Swaminathan's study on "Enhanced Compression of Test Data Employing Huffman Coding" published in the Journal of Theoretical and Applied Information Technology in May 2012.

[7]A.Al-Arabiya, S. Al-Hashemi, T. Khdour, M. Hjouj Btoush, S. Bani-Ahmad, and R. Al-Hashemi's research on "Lossless Image Compression Techniques through Combined Methods" published in the Journal of Software Engineering and Applications in 2012.

[8] Md. Rubaiyat Hasan's research on "Utilizing Huffman-based LZW Encoding Technique for Data Compression" published in the International Journal of Scientific & Engineering Research, Volume 2, Issue 11, November 2011.

[9] Monika and Pragati Kapoor's study on "Implementing Image Compression through a Hybrid Huffman-based LZW Approach" in 2007.

[10] A. Maliah, S. K. Shabbir and T. Subhashini's paper titled "Enhancing Image Compression and Quality through AnSpiht Algorithm with Huffman Encoder, Incorporating Retinex Algorithm" published in the June 2012 issue of the International Journal of Scientific & Technology Research, Volume 1, Issue 5.

[11] C. Saravanan and M. Surender's study titled "Improving Huffman Coding Efficiency with Lempel Zig Coding for Image Compression" published in the International Journal of Soft Computing and Engineering (IJSCE) in January 2013.

[12] Mr.D.V.Patil, Mr.S.G.SutarMrs.A.N.Mulla.. "Automated Improvement of Image Clarity through Retinex Method for Enhanced Visualization" published in the International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013.

[13]JiangXing-fanam) , "Enhancing Color Images with an Advanced Multi-Scale Retinex Algorithm" presented at the International Symposium on Photoelectronic Detection and Imaging in 2007 by Tao Chun-kan.

[14] Examining the correlation between image improvement and image size reduction within the framework of multi-scale Retinex. (Rahman, Z., Jobson, D.J., Woodall, G.A. 2011. Journal of Visual Communication and Image Representation, Volume 22, Pages 237–250, ScienceDirect)

[15]AsadollahShabhahrami, RaminBahrampour, MobinSabbaghiRostami, Mostafa AyoubiMobarhan, "Assessment of Huffman and Arithmetic Algorithms for Multimedia Compression Norms"