Journal of Electrical Engineering, Electronics, Control and Computer Science – JEEECCS, Volume 8, Issue 28, pages 1-6, 2022

A Fast Image Compression Web Service Using a REST API

Fidelis Odinma Chete and Alexander Omorokunwa
Department of Computer Science
University of Benin
Benin City, Nigeria

Abstract – **Images constitute a very important kind of data, and it forms an integral part of digital communication and other computer image processing applications. Due to their information-rich nature, images tend to require relatively large memory space to store them; and a lot of bandwidth to transmit them through a network. Compression makes it possible to reduce their file sizes while retaining a sufficient amount of quality that should still be pleasant to the human visual system. This compression can either utilize lossy or lossless compression techniques or methods. A web-based compression system which uses a smart lossy compression algorithm served through a REST API was presented in this work. The system proved efficient from the results obtained from testing with random large-size image files. In addition, acceptable compression ratio, compression time and space-saving percentages were derived in all the test cases.**

Keywords-compression, lossy, lossless, image processing, digital communication REST API

I. Introduction

With the rise of various means of generating digital images, there has also been a corresponding increase in the hardware and software cost of storing and moving (transmitting) them from one place to the other. Consequently, it has always been an active research space for computer scientists to devise effective and efficient techniques that can aid the compression of digital images; and also improve on already developed ones. This has led to the explosive growth of various industries that require the application of these technologies. Image compression is the foremost image processing capability that is utilized on the internet, satellite, military operations, digital (cable) television, medical and mobile communication. There has been tremendous improvement in the technology of image digitization. It now involves very simple processes to capture, store and transfer images. This has been made possible as a result of the fact that we now have very powerful stand-alone digital cameras and those embedded in mobile devices, like phones, with very fast processing capabilities and memory space. They produce images that are processed immediately. These images, developed from digital cameras, are usually large because of the sampling and quantization of light intensity that is used in creating the digital images. Thus, the process of transmitting or sharing these images would become a major obstacle and impractical. The preservation of the image (raw data) for future processing, considering the size of the storage facility and transmission media is an additional challenge. A method of solving this is by efficiently reducing the size of images, for easy storage and transmission. In addition, since digital images contain much visually irrelevant information, compressing it by removing this redundancy will be an effective method in maximizing storage capacity and increasing the speed of transferring it through a network.

Images contain relatively large volumes of data when compared to text and audio, thus the compression of images can be integrated into applications where several large images have to be archived in a limited storage space or situations where digital images are transmitted over limited channels. Consequently, two major factors which necessitate the compression of digital images are storage space and transmission (or bandwidth) cost. First, efficient storage utilization and archiving deals with ensuring that images are stored on devices with minimal file size but still retaining maximum image quality. Secondly, valuable computer networking resources (bandwidth and time) are utilized in transferring/transmitting digital images with large file sizes from one system to another. This is an additional cost incurred by businesses or organizations requiring image processing. An image compression framework can reduce these wastages or cost of hardware resources in digital image storage and transmission.

This study presents a web-based image compression system that can significantly reduce the image sizes of digital images (while retaining the essential features) and consequently resulting in the reduction of storage, transmission and computation costs. The research is limited to activities leading to the planning, analysis, design and development of an image compression application and then evaluating its performance using a batch of twenty (20) images with different attributes.

II. Review of Relevant Literature

Numerous compression algorithms have been developed over the years [1] and can be broadly categorized into two main classes; Lossy compression and Lossless compression.

Lossy compression is the most popular type of compression system [2] and produces images that are not exactly identical to the original image file because some image details (or information) are “lost” in the process.

[3] proposed that lossy image compression techniques should be mostly applied to natural or photographic images, since they will have little sharp edges in tone and gradient. This is because lossy compression leads to an acceptable loss of varied amount of image details or information.

According to [4], lossy compression techniques take advantage of the human visual systems (the eye) limitation in perceiving differences in colour details of an image (for example: very dark and bright tones.) In addition, the process typically achieves an approximate space saving ratio of about 1:10.

[5] explained that lossy compression is relatively more complex in utilizing computing resources than lossless methods, while [6] presented lossy compression methods as including (a) Transform coding (b) Vector Quantization and (c) Fractal coding.

In lossless compression there is no “loss” of data (information) contained in digital images [7]. Thus, if an image resulting from the compression method is decompressed it would be an exact replica of the original image file. When an identical reproduction of digital image is desired, it is recommended that lossless compression be used. To achieve this, digital images are broken down into smaller components that can be reassembled into the original image with no loss in information [8]. Lossless compression produces a relatively lesser compression ratio (2:1), since the quality of the image has to be retained after compression [9]. Lossless compression methods vary based on the type of data they are meant to compress – either text or data files. Since the quality of images are preserved in lossless compression, it can be applied to fields where there is need for high performance images like medical imaging, astronomy, geo-sensing among others [1]. There are various lossless methods that are used in compressing images, common ones include: Run-length encoding, Huffman encoding, LZW encoding Arithmetic encoding, and SCZ coding [10]

[1] posited that lossless compression is an essential tool in the process of storing and transmitting of medical images like MRI, CT and X-rays. This is based on the fact that since high quality images are required to make sound medical decisions, only minimal loss of image details can be entertained.

[7] proposed a simple modified fast Haar Wavelet transform method for image smoothing that proves lossless and its effects can easily be reversed to the original.

According to [3] lossless image compression preserves image information. If an image compressed with this method is decompressed, it would be an exact replica of the original image file. Consequently, lossless compression is recommended for images with well-defined line components like clip arts, technical drawings, etc.

[11] Surveyed various techniques of image compression and identified some popular lossless techniques like (a) Run-length Encoding (b) Huffman Encoding (c) Arithmetic Encoding and (d) LZW coding.

III. The Proposed System

A Web-based Image Compression System (WICS) is developed with the integration of an application programming back-end. The WICS is presented in an easily, accessible, efficient, cost-effective manner, with the goal of reducing the time and resources required to store and transmit digital images on the web. It will have the following features;

- ✧ The user uploads an image file from the computer to the web server (using a web browser).
- ✧ The system compresses the uploaded image file, on the back-end.
- ✧ The user downloads the compressed image from the server to his/her computer.

A. Functional Requirements

Functional requirements are a set of rules used to define what is expected of the system and its components. The functional requirements of the Image compression system are as follows:

- ✧ The system should be able to accept a digital image file from a user (uploading),
- ✧ The system will compress (reduce the size of) the image and store it in the web server,
- ✧ The system will generate download link for the compressed image so users can download.
- ✧ This system shall work with internet connectivity.

B. Non-Functional Requirements

The features of the proposed system include:

- **Increased effectiveness and efficiency:** a web-based image compression system (WICS) would enhance the storage and transmission of digital images by faster process of compression. More images will be compressed, thus leading to increased image compression within a given time frame.
- **Cost reduction:** a drastic reduction in the sizes of image would also lead to a reduction of storage and transmission costs
- **Cross-platform:** since it is a web-based system, the user would not need to install the application on their respective devices. The application can be accessed from any internet-enabled device (phones, tablets, PC) anywhere and any time.
- **Extensibility.** It can easily be integrated into other third-party applications that need a compression service, thereby preventing

reinventing-the-wheel and also serving as a valuable resource for other organizations.

C. *Components of the System*

The proposed system (WICS) is an automated web service that would ensure fast compression capabilities. This WICS architecture is a client-server model that has layers, namely: The Presentation layer, The Logic layer, and The Service layer.

- **The Presentation layer:**

This layer offers an interface with which the users can directly interact with the application. The prominent item in this layer is the web browser through which image files are sent (uploaded), and after compression received (downloaded) from the system.

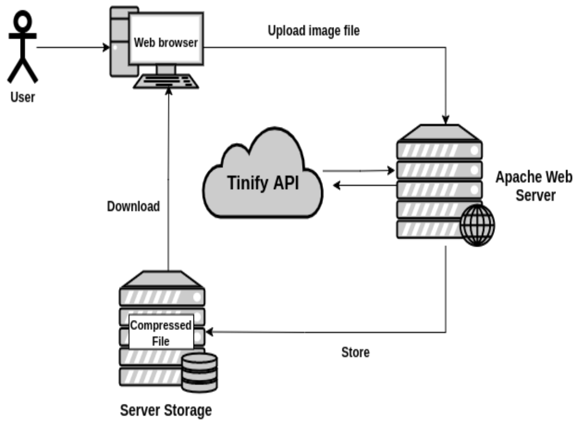
- **The Service layer:**

This is where the communication logic and controller logic are implemented. It determines what properties of image files are allowed to gain access to the system. This layer contains the web server and the server storage. The web server handles the users' requests or inputs (uploaded image file) and responds with an output (compressed image file). The server storage acts as the temporary repository of compressed image files to be downloaded by the user.

- **The Application layer:**

This layer contains the Application Programming Interface (API) engine which is responsible for compressing (processing) the image files uploaded to the server by the user and returning a compressed image to the server. These components are optimally integrated and interlocked to form the entire system architecture. This architecture is depicted in Figure 1.

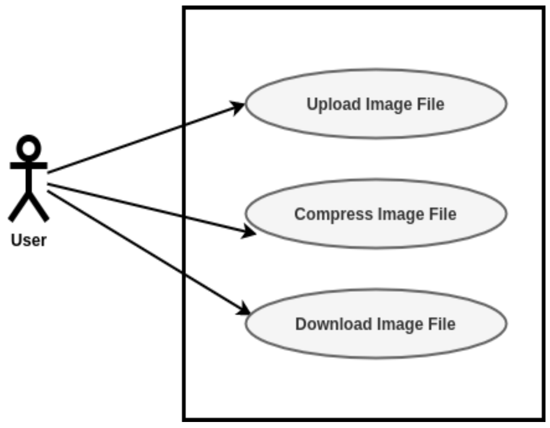
Figure 1. System Architecture



IV. USE-CASE DIAGRAM

Use cases for the system are identified to be "*upload image file*", "*compress image file*" and "*download image file*". The user uploads an image file from the computer. If the file is a valid image file, the system compresses the image. The system stores the compressed image file in a web server. The system displays a download link to the compressed file. The user downloads the compressed file. This is illustrated in figure 2

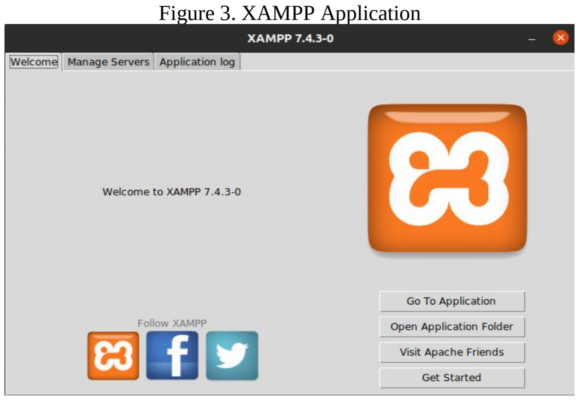
Figure 2. System Use-Case Diagram



Implementation architecture:

a) **The XAMPP Deployment Stack:** The web-based image compression application is deployed on a local server infrastructure called XAMPP. XAMPP stack is an open-source distribution of the PHP development environment. This XAMPP environment is very essential in creating web applications that run on servers (dynamic) using the PHP programming language. This is depicted in figure 3 and figure 4.

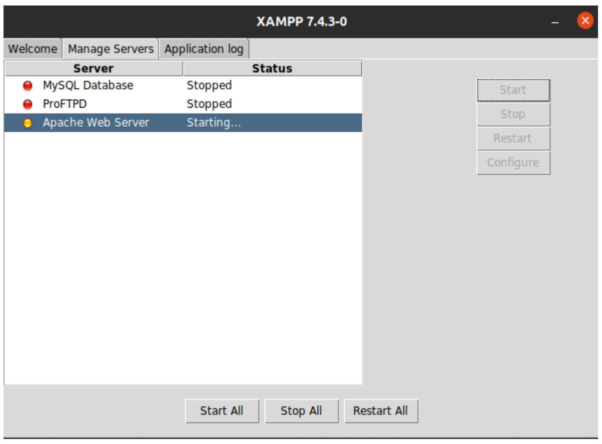
Figure 3. XAMPP Application



It consists of the following components; The Operating system, Linux (X),; Apache Web server (A),;MariaDB database (M);; PHP (P), and Perl (P). To make the setting up of a suitable programming environment for this work easy and straightforward, the XAMPP application was downloaded and installed on the test machine.

A summary of the entire implementation architecture (software and hardware) is summarized in Table 1 and Table 2 respectively.

Figure 4. XAMPP Server Management Dashboard



Server	Status
MySQL Database	Stopped
ProFTPD	Stopped
Apache Web Server	Starting...

TABLE I. SOFTWARE ARCHITECTURE

Operating System	Ubuntu Linux 20.04.1 LTS (64bit)
Server	XAMPP 7.4.3-0
Package Manager	Composer 1.8.6
API	Tinify 1.5
User Interface	Bootstrap 4.0.0
Web Browser (s)	Mozilla Firefox, Opera, Brave

TABLE 2. HARDWARE ARCHITECTURE

Memory (RAM)	4GB
Processor	Intel Celeron CPU N29402, 1.83GHz × 4
Hard Drive	500GB

V. GRAPHICAL USER INTERFACE

This section outlines some of the user interface results of the interaction by users with the system. They are presented along with screen-shots.

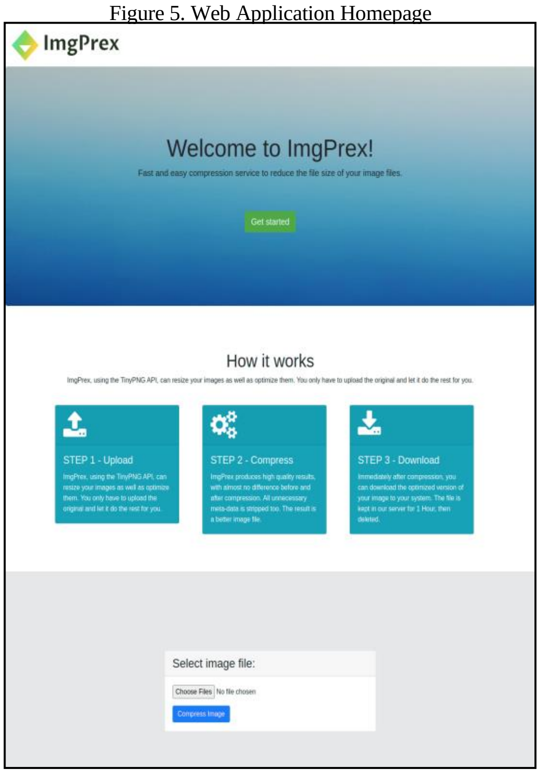
A. Home Page

The user starts the application by loading the home page in a web browser as depicted in figure 5. The user is presented with an upload form field through which an image file can be selected and uploaded from the computer device. When a file is selected, the user can then proceed to click on the "Compress Image" button to begin the compression of the uploaded file.

B. Download Page

If the user has uploaded the appropriate image file, the system delays for a variable amount of time (usually in a couple of seconds). Then a dialogue is displayed containing a successful message (that the file has been compressed), the time in seconds it took for the compression, and a download link to the compressed image file. The page also contains a navigation button that links to the home page where the user can repeat the process; else quit the application. This is depicted in figure 6.

Figure 5. Web Application Homepage



Welcome to ImgPrex!

Fast and easy compression service to reduce the file size of your image files.

How it works

ImgPrex, using the TinyPNG API, can resize your images as well as optimize them. You only have to upload the original and let it do the rest for you.

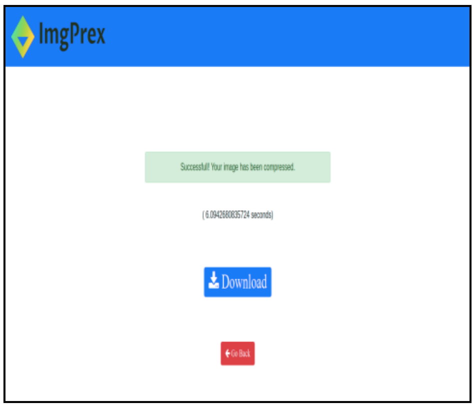
STEP 1 - Upload: ImgPrex, using the TinyPNG API, can resize your images as well as optimize them. You only have to upload the original and let it do the rest for you.

STEP 2 - Compress: ImgPrex produces high quality results, with almost no difference before and after compression. All unnecessary meta-data is stripped too. The result is a better image file.

STEP 3 - Download: Immediately after compression, you can download the optimized version of your image to your system. The file is kept in our server for 1 Hour, then deleted.

Select image file: No file chosen

Figure 6. Download Page



Successful! Your image has been compressed.

(6.09426808035724 seconds)

VI. RESULTS AND DISCUSSIONS

Some tests were conducted on the integrated compression system to evaluate its compliance with the specified functional requirements with the different parameters for measurement available. The study focused on three (3) common and useful parameters: compression ratio, space-saving, and compression time.

A. **Compression Ratio (CR):** This is the ratio between the compressed image file size and that of the uncompressed file (i.e., the original file).

$$\mathrm{CR} = \frac{\text{Uncompressed size}}{\text{Compressed size}}$$

B. **Space Saving (SS):** This parameter is expressed mathematically as the reduction in size of the compressed image relative to that of the original image file. In other words, the amount of redundant image data that was removed from the original uncompressed image file. It gives how effectively the application was able to save storage space. The usual way of representing this parameter is in the form of percentage. The mathematical expression to calculate space-saving is given as;

$$SS = 1 - \frac{\text{Compressed Size}}{\text{Uncompressed Size}}$$

C. **Compression Time (CT):** Compression time is the time it takes the application to reduce the file size of an uploaded image. Though the time of compression is important, getting an objective value is always very difficult. This is because the speed of most compression algorithms and applications depends on the implementation architecture (software and hardware) of the test machine. Thus, a faster machine would take lesser time to compress image files. Nonetheless, it was included in the testing to show that the application can run within relatively faster time frames.

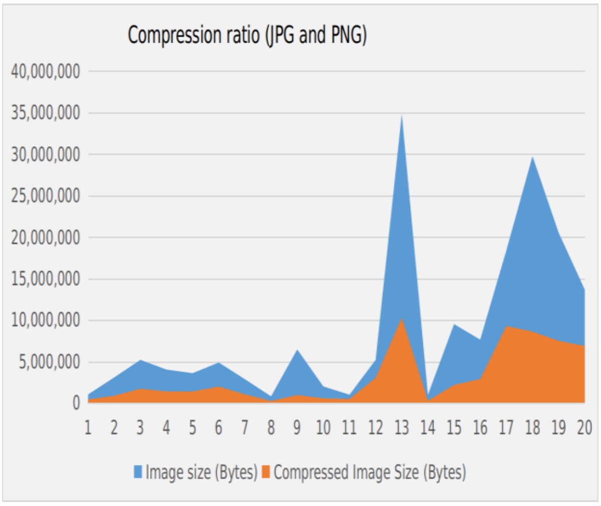
D. **Test Results:** The system was tested with twenty (20) randomly selected high-quality images (JPG and PNG) gotten from the internet. In all the test cases, the image files were successfully and efficiently compressed and the measurement parameters, like compression ratio and space saving were calculated in comparison with the original images. The result of this test is highlighted in Table 3 and Figures7-9.

TABLE 3. TEST RESULTS

Image S/N	Image Type	Image size (Bytes)	Compressed Image Size (Bytes)	Compression Time (sec)	Compression Ratio	Space Saving (%)
1	JPG	1,037,538	426,292	4.97	2.43	0.589
2	JPG	3,089,885	882,633	15.58	3.50	0.714
3	JPG	5,201,268	1,719,152	9.48	3.02	0.669
4	JPG	4,029,920	1,398,354	21.23	2.88	0.653
5	JPG	3,598,881	1,391,342	18.61	2.59	0.613
6	JPG	4,881,646	1,986,799	25.65	2.46	0.593
7	JPG	2,868,895	1,051,377	19.52	2.73	0.634
8	JPG	806,357	247,618	8.65	3.26	0.693
9	JPG	6,459,016	958,776	6.95	6.74	0.852
10	JPG	1,999,214	569,607	12.63	3.51	0.715
11	PNG	1,004,952	490,512	3.63	2.05	0.512
12	PNG	5,202,711	3,043,889	7.93	1.71	0.415
13	PNG	34,828,084	10,281,661	27.29	3.39	0.705
14	PNG	984,936	272,163	6.38	3.62	0.724
15	PNG	9,510,854	2,175,329	11.83	4.37	0.771
16	PNG	7,637,793	2,890,145	12.85	2.64	0.622
17	PNG	18,384,457	9,285,088	20.70	1.98	0.495
18	PNG	29,742,611	8,603,355	24.80	3.46	0.711
19	PNG	20,522,698	7,499,243	16.61	2.74	0.635
20	PNG	13,666,087	6,871,944	12.98	1.99	0.497

The results in Table 3 shows that the proposed system performed as expected. All the compressed image file sizes were efficiently reduced and were lesser than the original file sizes.

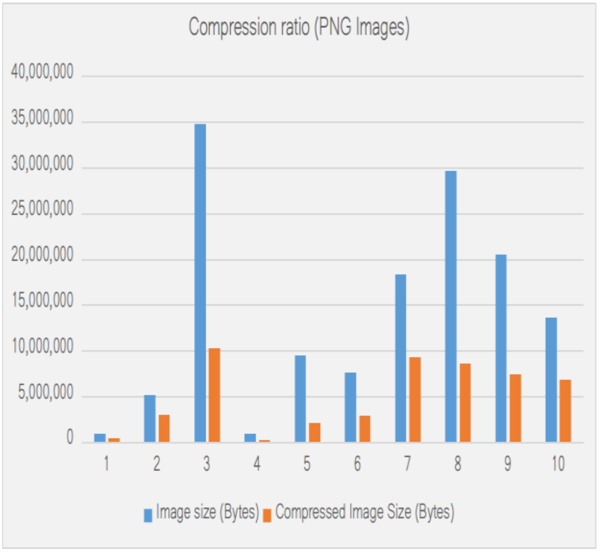
Figure 7. Compression Ratio (CR) of Test Images



The compression times were relatively faster; the slowest time was 27.29 seconds and the fastest was 3.63 seconds.

E. PNG IMAGES

Figure 8. Compression Ratio (CR) of Test PNG Images



The ten (10) PNG images compressed showed remarkable results. The largest space-saving was 77.1% and the smallest was 41.5%. In addition, the fastest compression time was 3.63 seconds and the slowest was 27.29 seconds.

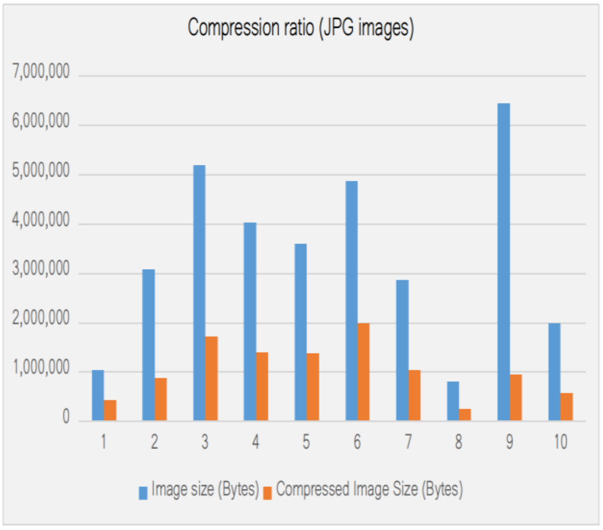
F. JPG IMAGES

JPG images tend to have lesser file sizes compared to PNG images. The ten (10) JPG images compressed also showed remarkable results. The largest space-saving was 71.5% and the smallest was 58.9%. In addition, the fastest compression time was 4.97 seconds and the slowest was 25.65 seconds.

G. JPG IMAGES

These results reveal that the Web-based Image Compression System met the required specifications.

Figure 9. Compression Ratio (CR) of Test JPG Images



CONCLUSION

Images are among the major forms in which content is being utilized and transferred in today's digital world. It is thus, imperative to have an easy-to-use and accessible compression system to enhance efficiency in utilizing and transmitting these digital images. In this study a simple and fast web-based compression method was developed that uses an available API framework to achieve remarkably high compression ratio. The system is made up of two parts the front-end (through which the user uploads a digital image from the computer) and the image compression back-end. The image compression system minimizes adequately the image file sizes while keeping a sufficiently desirable quality. Results demonstrated the efficiency of the proposed compression method on common digital image formats like JPG and PNG; which makes the system very valuable.

REFERENCES

[1] S. Jain, A Mittal, and S Roy (2011). Model-based image compression framework for CT and MRI images. International Journal of Medical Engineering and Informatics, 3(1).

[2] I. Mohammad and A. Zeekry (2015). Implementing Lossy Compression Technique for Video Codecs. International Journal of Computer Applications, 13(17), 44-51.

[3] K Arora and M. Shukla, (2014). A Comprehensive Review of Image Compression Techniques. International Journal of Computer Science and Information Technologies, 5 (2), 1169-1172.

[4] L.K. Tan (2006). Image File Formats. Biomedical Imaging and Intervention Journal, 2(1).

[5] S. Thakur and S. Rai, (2018). A Study Image Compression Techniques for the Number of Applications. International Journal of Research and Innovation in Applied Science, 3(4).

[6] R Goyal and J. Jaura, (2014) A Review of Various Image Compression Techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 4(7).

[7] Y.V. Balaso, S.S., Shinde & S.S. Tamboli (2016). Image compression using modified fast HAAR wavelet Transform. International Journal of Engineering Sciences & Research Technology, 5(8), 141-147.

[8] R. C. Gonzales and R.E. Woods, (2008). Digital Image Processing (3rd Ed). Addison-Wesley.

[9] M.I. Pu (2006). Fundamental Data Compression. Butterworth-Heinemann. London.

[10] D. Salomon (2008) A Concise Introduction to Data Compression: Undergraduate Topics in Computer Science. Springer-Verlag London Limited.

[11] G. Vijayvargiya, S Silakari and R. Pandey (2103) A Survey: Various Techniques of Image Compression. International Journal of Computer Science and Information Security, 11(10).