

# ***MAJOR MID-EVAL. REPORT FILE***



## **CRYPTAGE**

*Project 2017-2018*

***DEPARTMENT OF COMPUTER  
SCIENCE & IT***

### **Group Members:**

*Love Basant Kuntal (14103245)*

*Abhishek Patra (14103250)*

*Jagendra Singh (14103136)*

# **CONTENT**

## **1. Introduction:**

- 1.1 General Introduction,
- 1.2 Current/open problems
- 1.3 Problem Statement
- 1.4 Solution Approach.

## **2. Literature Survey**

Integrated summary of the referred papers

## **3. Analysis, Design and Modelling**

- 3.1 Functional requirements
- 3.2 Non Functional requirements
- 3.3 Overall architecture
- 3.4 Proposed Algorithm
- 3.5 Test Plan
- 3.6 **Current** Implementation

## **4. Appendix**

- 4.1 Project Plan as Gantt chart or WBS
- 4.2 Details of practice with new tool/technology (max 2 pages)
- 4.3 References

# **Introduction**

## **1.1 GENERAL INTRODUCTION**

Data compression is a process by which a file (Text, Audio, and Video) may be transformed to another (compressed) file, such that the original file may be fully recovered from the original file without any loss of actual information. This process may be useful if one wants to save the storage space. For example if one wants to store a 4MB file, it may be preferable to first compress it to a smaller size to save the storage space.

Also compressed files are much more easily exchanged over the internet since they upload and download much faster. We require the ability to reconstitute the original file from the compressed version at any time. Data compression is a method of encoding rules that allows substantial reduction in the total number of bits to store or transmit a file. The more information being dealt with, the more it costs in terms of storage and transmission costs. In short, Data Compression is the process of encoding data to fewer bits than the original representation so that it takes less storage space and less transmission time while communicating over a network .

## **1.2 CURRENT/OPEN PROBLEMS**

As data compression of image in various formats like jpeg,png etc are compressed into small –sized file in the same format ,videos are compressed accordingly with the same procedure ,various documents like pdf , doc etc are compressed accordingly with the same procedure .But as such there is no common platform available for the compression of all types file wether it is an image ,video or some other document . Here come the use of our app in this scenario providing a common platform for the compression of data

Till now, only image compression part is successfully implemented in the project and therefore adding the functionality of the image compression to the project and also providing the service of maintaining the quality of compression to the user .But apart from this ,the functionality of video,file and music compression is also to be implemented.

### **1.3 PROBLEM STATEMENT**

Nowadays as we have to send large amounts of data for processing and analytical purposes image compression plays a key role. It significantly increases the transfer speed and also the quantity of data is also enhanced. But one has to be careful while compression so as to not degrade the quality that it becomes useless. The compression mainly should have minimum degradation of quality and maximum size reduction. This is precisely what we are trying to achieve with this app.

The main aim of the project is to achieve compression and reduce the size of the files in the desired suitable quality. There are mainly four types of compression that we are required to implement:-



Image Compression

Video Compression

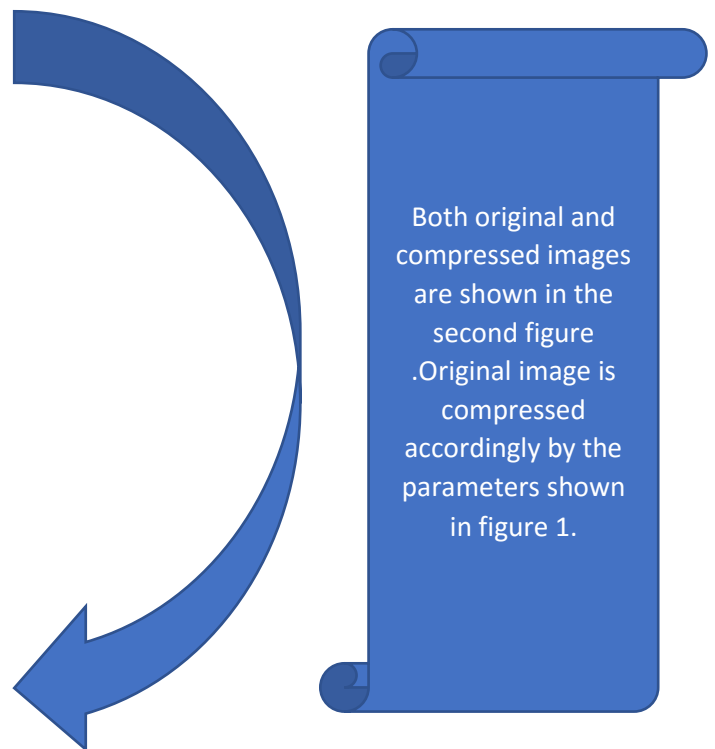
Zip Compression

Music Compression

Till now we have implemented one kind of compressions:-

Image Compression :- we convert the image in the different types according to the quality such as

10%,20%,50%,80% and zip compress.



## **1.4 SOLUTION APPROACH**

The solution to the defined problem statement is explained in the following steps:

- The user will provide the input in the of image ,video ,audio or any other file which is to be compressed .
- The obtained input from the user will then be passed to the level of compression to be made for maintaining the quality of the input.
- The output obtained through the compression will then be stored in the download folder in the internal storage of the device .
- The above output will be compressed and can easily be retrieved from the memory and can be stored in an efficient way in the device according to the desirable quality want by the user for efficient storage management.
- Further ,for the future implementation .we have also decided to add the functionality of encryption to the output compressed file so as to provide security or safety to the compressed file or output so that it can safely transmitter or shared over the network and can be reached out to the desired receiver.

# **Literature**

# **Survey**



- **A Study on Data Compression Using Huffman Coding Algorithms**

**D.Jasmine Shoba , Dr.S.Sivakumar ( Research Scholar , Assistant Professor) Department of Computer Science ,Department of Computer Applications ,Thanthai Hans Roever College, Perambalur Tamil Nadu – India**

**Introduction** : Data compression is one of the most widespread applications in computer technology. Algorithms and methods that are used depend strongly on the type of data, i.e. whether the data is static or dynamic, and on the content that can be any combination of text, images, numeric data or unrestricted binary data. The compression and decompression techniques are playing main role in the data transmission process. The best compression techniques among the three algorithms have to be analyzed to handle text data file. This analysis may be performed by comparing the measures of the compression and decompression

**Conclusion** : In this Paper Huffman coding compression techniques are compared. This system uses three metrics such as compression ratio, transmission time and memory utilization to compare and analyze the results. It is found that the Huffman coding technique shows the compression performance better techniques. Surely this technique will open a scope in the field of text compression where every bit of information is significant.

- **New Data Compression Algorithm and its Comparative Study with Existing Techniques**

**Rakesh Waghulde Information Technology Dept V.J.T.I., Mumbai – 400019**

**Harshal Gurjar Information Technology Dept. V.J.T.I., Mumbai – 400019**

**Vishal Dholakia Information Technology Dept V.J.T.I., Mumbai – 400019**

**G.P. Bhole Head of department Information Technology Dept V.J.T.I.,Mumbai – 400019**

**Introduction** : Data compression is a technique to represent data using less number of bits than original data. There are various data compression techniques available, but still there is a need to achieve more compression ratio. This paper proposes an algorithm that combines the features of both Huffman's algorithm and LZW algorithm to achieve more compression ratio. This algorithm is named as VJ Zip. In the new algorithm VJ Zip, for compression, firstly every duplicate occurrence of data is replaced with the pointer to its previous occurrence to obtain partially compressed data. From this partially compressed data, the literals and pointers are further compressed using two separate Huffman trees. We measure the performance of this new algorithm in terms of compression ratio and also compare the performance of this new modified algorithm with the two algorithms viz., Huffman's algorithm and LZW algorithm, individually. Comparing the results it is inferred that new modified algorithm, VJ Zip, is more efficient than Huffman's algorithm and LZW algorithm applied individually. On an Average, it achieves 26% & 54% more compression ratio for .txt and .xml format respectively, as compared to Huffman's algorithm and 16% & 18% more compression ratio for .txt and .xml format respectively, as compared to LZW. Also this paper compares the performance of new algorithm with the existing software 7Zip. As compared to 7Zip new modified algorithm gives almost same compression ratios for text format while achieves 1% more compression ratio for images and videos.

**Conclusion** : Huffman algorithm works on all the file formats which uses lossless compression. LZ77 and LZW are more effective on text file formats. Huffman algorithm and LZ77 algorithm are more efficient for file formats which can undergo lossless compression (.txt, .xml, .docx) than those with lossy compression (.jpeg, .gif, .mp3, .mp4). Proposed algorithm, VJ Zip, combines the feature of Huffman algorithm and LZW.

As compared to Huffman's algorithm, proposed algorithm achieves 26% & 54% more compression ratio for .txt and .xml format respectively. As compared to LZW algorithm, proposed algorithm achieves 16% & 18% more compression ratio for .txt and .xml format respectively. Results of proposed algorithm are best achieved on XML file format (.xml). An average of 91% compression ratio can be achieved for XML file format. For proposed algorithm, Repetition of data in file increases the compression ratio. Compression Techniques are more suitable for large data sizes.

- **SECOMPAX: A bitmap index compression algorithm**

**Introduction** : This work introduces a new bitmap encoding method called Scope-Extended COMPRESSED Adaptive index (SECOMPAX) which records the origin bitmap index into a new format according to a new designed codebook.

**Conclusion** : In this paper, we propose a new bitmap index compression algorithm named SECOMPAX (ScopeExtended

COMPRESSED Adaptive index), which considers the possibility where the number of 1's are comparable to 0's or even much more than 0's in the original bit sequences in bitmap index. Thus, when the occurrence of consecutive 0's is not extremely much, our experiments results show that SECOMPAX performs better compression ratio and faster querying speed compared with the state-of-art algorithm, i.e., WAH, PLWAH and COMPAX. What's more, the decoding time still remains comparable to the state-of-art algorithm.

- **Research Paper on Text Data Compression Algorithm using Hybrid Approach**

***Amandeep Singh Sidhu [M.Tech]<sup>1</sup>, Er. Meenakshi Garg [M.Tech]<sup>2</sup> C.S.E.  
& Guru Kashi University, Talwandi Sabo, Bathinda, Punjab, India***

**Introduction** : Data compression is a process by which a file (Text, Audio, and Video) may be transformed to another (compressed) file, such that the original file may be fully recovered from the original file without any loss of actual information. This process may be useful if one wants to save the storage space. For example if one wants to store a 4MB file, it may be preferable to first compress it to a smaller size to save the storage space

**Conclusion** : In this proposed work, an improved dynamic bit reduction algorithm is developed to compress and decompress the text data based on lossless data compression approach. Various experiments have been conducted on different datasets such as Random, Alphanumeric, Numeral and Special Characters dataset

**Analysis,**  
**Design**  
**and**  
**Modelling**

### **3.1 FUNCTIONAL REQUIREMENTS**

The functional requirements essentially specifies what a system must do to achieve a particular goal. Thus the functional requirements of this project includes:

- The user will provide the input in the of image ,video ,audio or any other file which is to be compressed .
- The obtained input from the user will then be passed to the level of compression to be made for maintaining the quality of the input.
- The output obtained through the compression will then be stored in the download folder in the internal storage of the device .
- The above output will be compressed and can easily be retrieved from the memory and can be stored in an efficient way in the device according to the desirable quality want by the user for efficient storage management.
- Further ,for the future implementation .we have also decided to add the functionality of encryption to the output compressed file so as to provide security or safety to the compressed file or output so that it can safely transmitter or shared over the network and can be reached out to the desired receiver.

Following are the files which will be implemented in the project:

- 1)Image Compression
- 2)Video Compression
- 3) Zip Compression
- 4)Music Compression

### **3.2 NON-FUNCTIONAL REQUIREMENTS**

Non- functional requirements include how the model proposed would work.

1. Error Handling: the system shall handle expected and non expected errors in ways that prevents loss in information
2. Performance Requirements: the system shall handle large number of users and response should take less amount.

3. Safety Requirements: system shall not cause any harm to the user.

### **3.3 OVERALL ARCHITECTURE/TOOLS**

#### **ANDROID :**

It will be used for providing a platform to the application for its execution involving the making of UI design of the app and adding for adding all activities and utilities to the application through ANDROID STUDIO.

#### **JAVA:**

This object oriented coding language will be used to implement the code in android studio for all the activities.

#### **Bitmap compression :**

It is one of the efficient techniques used for the purpose of data compression.

#### **Zip :**

It is the extension used for compressed files

### **3.4 PROPOSED ALGORITHM**

#### **Image Compression:-**

```
bitmap1.compress (Bitmap.CompressFormat.JPEG, i, bytearrayoutputs  
stream) ;
```

```
BYTE = bytearrayoutputstream.toByteArray();  
bitmap2 = BitmapFactory.decodeByteArray(BYTE, 0, BYTE.length) ;  
int w=bitmap2.getWidth();  
int h=bitmap2.getHeight();
```

The above is the main code used for image compression.

In this 'i' is the quality that we have to send to compress the image and it would be compressed in the jpeg format. The size of the compressed image varies according to the value of 'I' you pass by selecting the appropriate option.

### **3.5 TEST PLAN**

- **Compression ratio:** Compression ratio is defined as the ratio of size of the compressed file to the size of the source file.

$$\text{Compression ratio} = (C2/C1) * 100\%$$

- **Saving Percentage:** Saving Percentage calculates the shrinkage of the source file as a percentage.

$$\text{Saving percentage} = (C1 - C2/C1) * 100\%$$

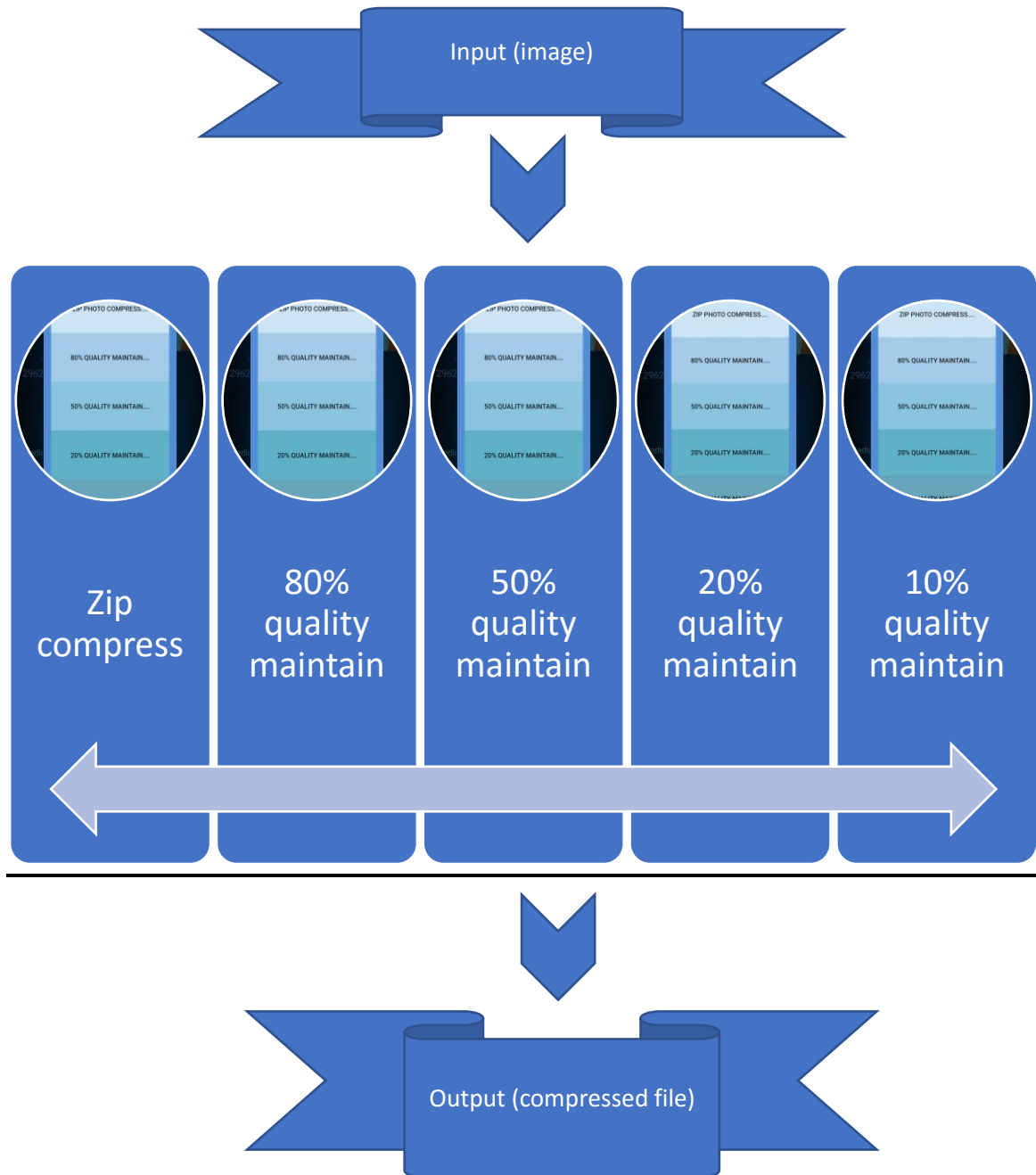
C1= Size before compression

C2= Size after compression

Below is the table obtained ,when a file of size 2962KB is compressed

<b>Input text size (in bytes) C1</b>	<b>Output of proposed System ( in bytes)(after 80,50,20,10%compression respectively)C2</b>	<b>Compression Ratio of proposed system (In %)</b>	<b>Saving Percentage</b>
2962	1275	43.04	56.95
2962	509	17.18	82.81
2962	267	9.01	90.98
2962	170	5.73	94.26

### **3.6 CURRENT IMPLEMENTATION**





# **Appendix**

## **4.1 PROJECT PLAN**

- The user will provide the input in the of image ,video ,audio or any other file which is to be compressed .
- The obtained input from the user will then be passed to the level of compression to be made for maintaining the quality of the input.
- The output obtained through the compression will then be stored in the download folder in the internal storage of the device .
- The above output will be compressed and can easily be retrieved from the memory and can be stored in an efficient way in the device according to the desirable quality want by the user for efficient storage management.
- Further ,for the future implementation .we have also decided to add the functionality of encryption to the output compressed file so as to provide security or safety to the compressed file or output so that it can safely transmitter or shared over the network and can be reached out to the desired receiver.

## **4.2 DETAILS OF PRACTICE WITH NEW TECHNOLOGY:**

We aim to achieve the compression of different files mainly including images ,videos ,audio etc on a single platform through different compression and encryption techniques.

We have chosen Android studio as the platform to build our project with help of JAVA language as the programming language.

## **4.3 REFERENCES**

- Research Paper on Text Data Compression Algorithm using Hybrid Approach  
Amandeep Singh Sidhu [M.Tech]1, Er. Meenakshi Garg
- New Data Compression Algorithm and its Comparative Study with Existing Techniques  
Rakesh Waghulde Information Technology Dept V.J.T.I., Mumbai – 400019  
Harshal Gurjar Information Technology Dept. V.J.T.I., Mumbai – 400019  
Vishal Dholakia Information Technology Dept V.J.T.I., Mumbai – 400019  
G.P. Bhole Head of department Information Technology Dept V.J.T.I., Mumbai – 400019
- SECOMPAX: A bitmap index compression algorithm
- A Study on Data Compression Using Huffman Coding Algorithms D.Jasmine Shoba.  
[1], Dr.S.Sivakumar [2] Research Scholar [1], Assistant Professor [2] Department of  
Computer Science [1] Department of Computer Applications [2] Thanthai Hans  
Roever College, Perambalur Tamil Nadu – India