Image Compressor

Submitted by

Yogesh Triveni Gupta

2067400

April 2018

Under the guidance of

Dr. HIREN DAND

in partial fulfillment of the requirements for qualifying

B.Sc.-(I.T.), Semester – VI Examination

Parle Tilak Vidyalaya Association's

MULUND COLLEGE OF COMMERCE

(Affiliated to University of Mumbai)

(NAAC "A" Grade Re-accredited 2016 - 2021)

S. N. ROAD, MULUND WEST, MUMBAI 400080

2017 - 2018

Parle Tilak Vidyalaya Association's

MULUND COLLEGE OF COMMERCE

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled **Image Compressor**

Undertaken at **PTVA's MULUND COLLEGE OF COMMERCE** *by* **Yogesh Triveni Gupta**

Seat no. **2067400** *in partial fulfilment of B.Sc. IT degree (Semester. VI) Examination has successfully completed the all the phases of the project under my supervision during the academic year 2017 - 2018.*

Internal Guide and Coordinator

External Examiner

Principal

Date: **23/04/2018**

College Seal

Table of Contents

Synopsis

Title:	Image Compressor
Description:	A windows software application "Image Compressor" to compress image(s) stored on the physical storage device. The application will facilitate the following two most important aspect: • Store data in an efficient form • Transmit data in an efficient form Some of the product features will be as follows: • Wieldy, Understandable and Interactive UI. • It will support all image file formats. • It will be able to work in background. • It will have signed assembly. • It will be able to resist "Reverse Engineering". • It will have a portable executable (.exe). • It will follow secure installation. Some of the product functions will be as follows: • Adding images: It will allow the user to be able to add image files in two ways ▪ One by one or multiple at a time ▪ From a folder containing images • Removing images: It will allow the user to be able to remove images in two ways ▪ One by one or multiple at a time ▪ All at a time • Setting target path: It will allow the user to be able to set the destination path for the images being compressed. • Selecting quality: It will allow the user to be able to select the quality in which compression to be done. • Working in background: The application will be able to work in background mode to complete its functionality.
Objective:	The main objective of "Image Compressor" is to facilitate the following:

	• Store data in an efficient form • Transmit data in an efficient form
Developer:	Yogesh Gupta
Programming Languages:	C#.Net
Hardware and Software For Development:	Hardware: • Toshiba Satellite C50-A I2011 ▪ Screen Size: 15.6 inches ▪ Processor: Intel Core i3 2348M ▪ RAM: 4 GB ▪ Hard Drive: 640 GB Software: • Windows 8.1 Pro 64-bit ▪ Visual Studio 2015 Community ▪ Confuser ▪ .Net Reflector ▪ Inno Setup
Hardware and Software For Playback:	Hardware: • PC or Laptop ▪ Screen Size: Any available ▪ Processor: Any available ▪ RAM: Any available ▪ Hard Drive: Any available Software: • Any windows operating system available
Deployment Platform	An web based platform from ware all the end users can download and get guided to use the software: www.4imagecompressor.blogspot.com

Software requirement specification

1. Introduction

Purpose:

- To describes the functional and non-functional requirements of the software product “Image Compressor” which is to be developed.
- To help developer(s) to implement and verify the correct functioning of the software product “Image Compressor”.

Project Scope and Product Features:

Project Scope:

- The software product “Image Compressor” will allow the user to compress images stored on the physical storage device.

Product Features:

- Wieldy, Understandable and Interactive UI.
- It will support all image file formats.
- It will be able to log exceptions in a text file.
- It will be able to work in background.
- It will have signed assembly.
- It will be able to resist “Reverse Engineering”.
- It will have a portable executable (.exe).
- It will follow secure installation.

2. Overall description

Product Perspective:

- “Image Compressor” will be a windows application that will solve one common problem that is “How to store large volume of image data in less space”.
- It will be a windows software application that will allow the user to compress image(s) stored on the physical storage device.

Product Functions:

- Adding images: It will allow the user to be able to add image files in two ways
  - One by one or multiple at a time
  - From a folder containing images
- Removing images: It will allow the user to be able to remove images in two ways
  - One by one or multiple at a time
  - All at a time
- Setting target path: It will allow the user to be able to set the destination path for the images being compressed.
- Selecting quality: It will allow the user to be able to select the quality in which compression to be done.
- Logging exceptions: The application itself will be able to log exceptions in a text file when an uncaught exception is raised.
- Working in background: The application will be able to work in background mode to complete its functionality.

3. User Classes and Characteristics

- "Image Compressor" will be available for all the windows computer users, who would like to compress image files for any reason.
- Generally, "Image compressor" users can be divided into two categories
  - Ones who have already used such image compressing software: Such users will be able to use the software in its full optimization
  - Others who never used such software: UI is very simple to use and work with, which enables such users to use the common functionality of the application. Such users can also learn to use the software in its full optimization such as removing unwanted images, selecting the right quality for the image etc. On the web portal www.4imagecompressor.blogspot.in
- Besides, there may be another category of users may exist, based on their computer knowledge.

- More knowledgeable and experienced users of the computer system: Such users can easily use the software of its best
- Less knowledgeable and experienced users of the computer system: Such users can simply google www.4imagecompressor.blogspot.in and go through our tutorial on how to download, install and use the software.

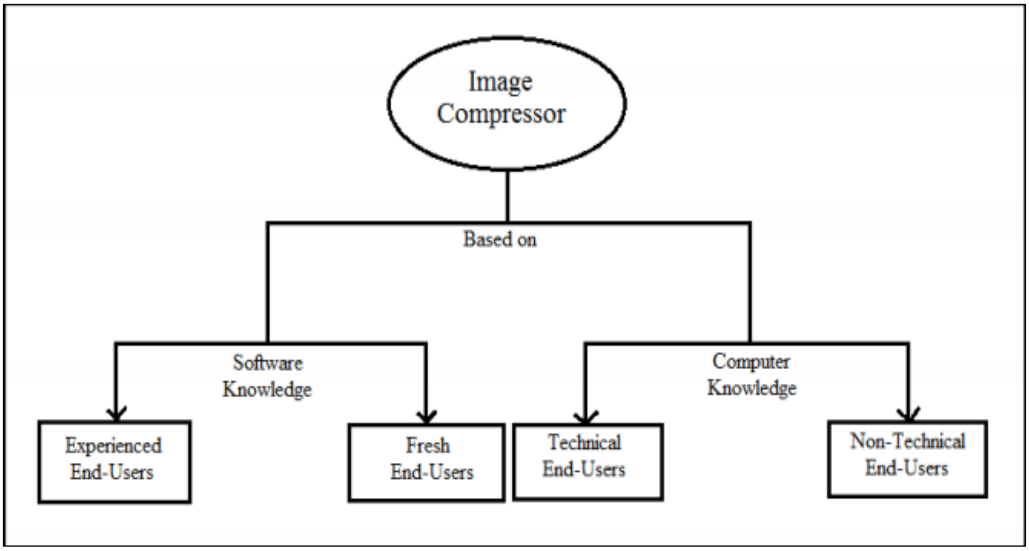


Figure 3.1: User classes and characteristics

4. Operating Environment

- "Image Compressor" will be able to work in a computer or laptop with the following hardware and software components.
  - Hardware Components:
    - ✓ Any available CPU
    - ✓ At least 100MB of free hard disk of any available size
    - ✓ Any available size of RAM
    - ✓ Any available resolution of display
  - Software Components
    - ✓ Windows 7 / 8 / 8.1 / 10
    - ✓ .NET Framework 4.0 or above

5. Design and Implementation Constraints

- All UI design and the functionalities will be implemented in C#, hence working knowledge of C# programming language will be mandatory.
- Use of Visual Studio software of any version and edition will be mandatory with the following components
    - C# Language Dependencies
    - .Net framework

6. User Documentation

- Users can google www.4imagecompressor.blogspot.in for the following queries
    - How to download the software?
    - How to install the software?
    - How to use the software?

7. External Interface Requirements

User Interface:

- The “Image Compressor” which is to be developed, will be based GUI Interface.
- It will be having three main sections with their subsections as described bellow
    - Tool Bar
        - ✓ Add Files
        - ✓ Add Folder
        - ✓ Remove Selected
        - ✓ Clear All
        - ✓ Target
        - ✓ Compress
    - Display Area
        - ✓ File Name
        - ✓ Extension

- ✓ Size
- ✓ New Size
- ✓ Compression %
- ✓ Status
- ✓ Source Path
- ✓ Status Bar
- ✓ Files Count
- ✓ File Processed
- ✓ File Reaming

- It will be having both text based indication and image symbols for each of the options available to make it easy to understand and use, also making sure that a non-technical person even can use it.
- The above hierarchy of GUI and their functionalities will be developed using C# programming language.

Hardware Interface:

- The "Image Compressor" will be an windows application and hence it can only communicate with computes and laptops with the following hardware
  - At least 50MB free hard disk of any available size
  - Any available size of RAM
  - Any available resolution of display

Software Interface:

- "Image Compressor" is a windows application and hence it needs to be operated with the following windows operating system of any edition along with the component specified.
  - Windows 7 / 8 / 8.1 / 10
  - .NET Framework 4 or above

8. System Features

Wieldy, Understandable and Interactive UI:

- Description and Priority

        - "Image Compressor" will have an easy to use, understandable and interactive UI so that even a non-technical person can use it.
        - Priority is high because, the users of this application may not be from technical background.
    - Stimulus/Response Sequence
        - Stimulus: User will click on one of the symbol.
        - Response: If the symbol which is clicked will not need any requirements to be filled it will perform its action else it pops a message indicating what needs to be done.
    - Functional Requirements
        - The application must have easy and understandable symbols to indicate specific functionality.
        - Users must have visible indication of all the operations during compression.

Support for all image file formats:

- Description and Priority
    - "Image Compressor" will be able to support all image file formats such as jpeg, jpg, jfif, exif, gif, png, bat, bmp, tiff.
    - Priority is high because, users may have their own choice of image file format to work with.
- Stimulus/Response Sequence
    - Stimulus: User will select image(s) to compress.
    - Response: application will take those images for compression.
- Functional Requirements
    - Cases to accept all image file formats must be specified in the application development logic.

Add images:

- Description and Priority
    - "Image Compressor" will allow the user to be able to add image files in two ways
        - ✓ One by one or multiple at a time

        - ✓ From a folder containing images
        - ✓ Priority is high because, without adding the image(s) further operations can't be done.
- Stimulus/Response Sequence
    - Stimulus: User will click to add one by one or multiple at a time.
    - Response: If the selected file(s) will be of type image file format, application will accept the file else it will discard the file(s).
    - If Open is clicked application will add the selected image files else it will cancel the selection.
    - Stimulus: User will click to select from a folder.
    - Response: If the selected folder will contain all file(s) of type image file format application will accept the file(s) else it will discard the file. If OK is clicked application will add the all acceptable image files from the selected folder else it will cancel the selection.
- Functional Requirements
    - Application must be able to make decision, which files have to be accepted and which not.
    - If nothing is added, application must be able to give appropriate message to the user.

Remove images:

- Description and Priority
    - "Image Compressor" will allow the user to be able to remove image files in two ways
        - ✓ One by one or multiple at a time
        - ✓ All at a time
    - Priority is medium because, not necessary that users will add unwanted image files.
- Stimulus/Response Sequence
    - Stimulus: User will select one or more image from the selected set and click to remove.
    - Response: the selected image will be removed.
- Functional Requirements

- Application must be able to allow user to select one or more image at a time.

Set target path:

- Description and Priority
  - "Image Compressor" will allow the user to set the destination path for the image(s) being compressed.
  - Priority is high because, application will not be able to set the default path for the image(s) being compressed.
- Stimulus/Response Sequence
  - Stimulus: User will click to set the target path.
  - Response: Application will pop up a form to take value from the user.
  - Stimulate: User will browse to select the target path.
  - Response: If Ok is clicked application will set the selected path as target path else it will cancel the selection.
  - Stimulation: User will choose to make a new directory on the selected path.
  - Response: If OK is clicked application will set the path of newly created directory as target path else it will cancel the selection.
- Functional Requirements
  - Application must be able to set the selected pat as target path for image(s) being compressed.
  - Application must be able to make a new directory on the selected path.

Select quality:

- Description and Priority
  - "Image Compressor" will allow the user to able to select the quality in which compression to be done.
  - Priority is medium because, even if the user will not select any quality it will select the default one.
- Stimulus/Response Sequence
  - Stimulus: User will click to set the image quality.

    - Response: Application will pop up a form to select value for quality from the user.
    - Stimulus: User will select a value for quality from the list.
    - Response: If Accept is clicked application will take the value for compression operation else it will take the default value.
- Functional Requirements
    - Application must be able to take the selected quality value for the compression operation.
    - Application must be able to take the default value if no value is selected.

Log exceptions:

- Description and Priority:
    - "Image compressor "will be able to log exceptions in a text file when an uncaught exception is raised.
    - Priority is medium because, this will not contribute to the compression operation but, this will help in development of further releases.
- Stimulus/Response Sequence
    - Stimulus: Considering, user will have done something which causes application to raise an exception.
    - Response: Application will create a log file and write the description of the exception raised.
- Functional Requirements
    - Application must be able to create a text file for logging exceptions.

Work in background:

- Description and Priority
    - "Image Compressor" will be able to work in background mode to complete its functionality.
    - Priority is medium because, user may use to run the software alone without forcing it to run in background.
- Stimulus/Response Sequence

    - Stimulus: User will minimize the image compression window to work on other application.
    - Response: If it is performing compression operation it goes into background mode to compete the operation else it will just minimize itself.
- Functional Requirements
    - Application must be able to complete its compression operation in background.

Signed assembly:

- Description and Priority
    - “Image Compressor” will have a private key to secure the generation and deployment of next version of the same application.
    - Priority is low because, this will not contribute to the compression operation but, this will help to make a new version of the same application securely.
- Stimulus/Response Sequence
    - Stimulus: Any developer with the intension of generating or deploying the new version of the same application will try to work for it.
    - Response: Application will ask for the private key to work on it, if provided it will allow to work on it else it will not allow to work on it.
- Functional Requirements
    - Developer must be able to sign the assembly and generate the private key for the software application.

Resist reverse engineering:

- Description and Priority
    - “Image Compressor” will be able to resist reverse engineering.
    - Priority is high because, others can misuse the same logic of development for their own financial gain.
- Stimulus/Response Sequence

    - Stimulus: Any person with the intension of getting the source code or application logic of the application will bypass the software through the software like .NET Reflector.
    - Response: Application will hide all of its source code and application logic.
- Functional Requirements
    - Developer of the application must be able to make application resistible to reverse engineering.

Portable executable:

- Description and Priority
    - “Image Compressor” will have a portable executable so that both 64 bit (x64) and 32 bit (x86) operating system users can install and use the software.
    - Priority is high because, this application will be made by the vision that all windows operating system users can make use of it.
- Stimulus/Response Sequence
    - Stimulus: A 64 bit (x64) or 32 bit (x86) windows operating system user will try to install the application.
    - Response: If .NET Framework 4 or above is already installed application will be installed properly and user can use it else application will be installed but, it will only be usable once .NET Framework 4 or above is installed.
- Functional Requirements
    - Developer must be able to generate a portable executable of the software application.

Secure installation:

- Description and Priority
    - “Image Compressor” installer will be protected with a predefined password to complete the installation.

    - Priority is low because, this will not contribute to the compression operation but, this will help to make sure a legitimate user is trying to use the software application.
- Stimulus/Response Sequence
    - Stimulus: Use will try to install the software via portable executable (.exe) or installer.
    - Response: In the progress, a form will ask for a password to complete the installation. Stimulus: User will enter the password
    - Response: If the entered password will be the same as that of set by the developer installation will progress and complete else installer will show the appropriate message and end the installation progress.
- Functional Requirements
    - Developer of this application must set a predefined password for its complete installation.

9. Other Non-Functional Requirements

Performance Requirements:

- UI of “Image Compressor” must be designed so that it will be compatible with all display resolutions.
- Adding, removing, setting target path and selecting image quality must not take to log to process.
- It must be able to process all the image files selected by the user at that movement.
- Application must not take to log to process or compress the image files, may be around 100 images in 1 min will be acceptable.
- Logging of exceptions must not result in failure of the application and user may not be able to recognize the exception logging process.
- It must be able to work same as expected in background as well.
- Portable executable must be able to work in the same way for both 64 bit (x64) and 32 bit (x86) processor architecture.
- Secure installation must not affect the complete installation thus, functioning of the software application.

Security Requirements:

- The complete installation of software will require a predefined password set by the developer.
- The application will have two types of security
    - Signing the Assembly: This will ensure that no one can generate a new version of the application without having the private key
    - Code Obfuscation: This will prevent the application form reverse engineering.

Software Quality Attributes:

Adaptability:

- The application will be able to adjust itself for different screen resolutions.
- It will adopt itself for large number of images in the same display area making the area scrollable.

Availability:

- The application will be able to serve its purpose on any time once it is installed.

Flexibility:

- The display area in the application will be resizable to the user.

Portability:

- The application will be portable to both 64 bit (x64) and 32 bit (x86) windows operating system.

Reliability:

- The application will be reliable to work with till the time it is not infected by any viruses.

Usability:

- The application will be able to process as many images as user wants.

Objective and scope of the project

1. Objective of the project

Image compression plays very important role in storing and transferring images efficiently. With the evolution of electronic devices like Digital Cameras, Smart phones, biometric applications, the requirement of storing images in less memory is becoming necessary. Using image compressing software the requirement of less memory storage and efficient transmission can be fulfilled. At the same time it should be considered that, the compressing software must be able to compress the image with efficient loss as per the user requirement. The objective of the project is to have such compressing software which will allow user to select the level of compression he or she wishes to perform on the image.

The increasing demand for multimedia content such as digital images has led to great interest in research into compression tools. The development of higher quality and less expensive image acquisition devices has produced steady increases in both image size and resolution, and a greater consequent for the design of efficient compression tools. Although storage capacity and transfer bandwidth has grown accordingly in recent years, many applications still require compression.

In general, this project investigates all image file formats and the compression tools available and building a simple image compressing software. The main objective is to design and develop compressing software for windows operating system to make efficient use of storage devises and bandwidth available in the transmission medium. Factors related to the need for image compression include:

- The large storage requirements for multimedia data
- Effective use of the transmission medium
- Low power devices such as handheld phones have small storage capacity
- Network bandwidths currently available for transmission

Hence, I can conclude, the main objective of my project is to facilitate the following:

- Store data in an efficient form
- Transmit data in an efficient form

2. Scope of the project

The project “Image Compressor” is a windows application, hence it will be available for all the windows computer users, who would like to compress image files for any reason, which will allow the user to compress images stored on the physical storage device with the following functionalities.

- Adding images: It will allow the user to be able to add image files in two ways
  - ✓ One by one or multiple at a time
  - ✓ From a folder containing images
- Removing images: It will allow the user to be able to remove images in two ways
  - ✓ One by one or multiple at a time
  - ✓ All at a time
- Setting target path: It will allow the user to be able to set the destination path for the images being compressed.
- Selecting quality: It will allow the user to be able to select the quality in which compression to be done.
- Working in background: The application will be able to work in background mode to complete its functionality.

Generally, the users can be divided into two categories ones who have already used such image compressing software and others who never used such software. Besides, there may be another category of users may exist, based on their computer knowledge such as more knowledgeable and experienced users of the computer system and less knowledgeable and experienced users of the computer system. All the user can have benefit of the software via tutorial on our web portal www.4imagecompressor.blogspot.in

Besides, the project has some dependencies associated with it for both development and use. It uses Visual Studio Community 2015 software with C# language dependencies and .NET framework 4.0 for complete building of the software. Hence, the software product can only be used when we have .NET framework installed in our windows operating system.

Theoretical background

1. Image Compression

The increasing demand for multimedia content such as digital images and video has led to great interest in research into compression techniques. The development of higher quality and less expensive image acquisition devices has produced steady increases in both image size and resolution, and a greater consequent for the design of efficient compression systems. Although storage capacity and transfer bandwidth has grown accordingly in recent years, many applications still require compression.

In general, this thesis investigates still image compression in the transform domain. Multidimensional, multispectral and volumetric digital images are the main topics for analysis. The main objective is to design a compression system suitable for processing, storage and transmission, as well as providing acceptable computational complexity suitable for practical implementation. The basic rule of compression is to reduce the numbers of bits needed to represent an image. In a computer an image is represented as an array of numbers, integers to be more specific, that is called a "digital image".

- The large storage requirements for multimedia data
- Network bandwidths currently available for transmission

Figure 1.1: Image Compression

2. Data Compression Model

A data compression system mainly consists of three major steps and that are removal or reduction in data redundancy, reduction in entropy, and entropy encoding. A typical data compression system can be labelled using the block diagram shown in Figure 4.2 It is performed in steps such as image transformation, quantization and entropy coding. JPEG is one of the most used image compression standard which uses discrete cosine transform (DCT) to transform the image from spatial to frequency domain. An image contains low visual information in its high frequencies for which heavy quantization can be done in order to reduce the size in the transformed representation. Entropy coding follows to further reduce the redundancy in the transformed and quantized image data.

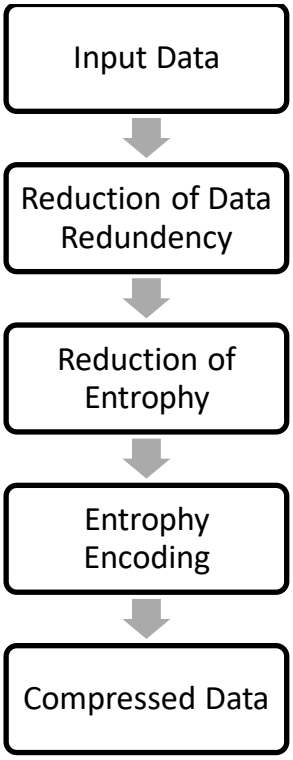


Figure 2.1: Data Compression Model

The main advantage of compression is that it reduces the data storage requirements. It also offers an attractive approach to reduce the communication cost in transmitting high volumes of data over long-haul links via higher effective utilization of the available bandwidth in the data links. This significantly aids in reducing the cost of communication due to the data rate reduction. Due to the data rate reduction, data compression also increases the quality of multimedia presentation

through limited-bandwidth communication channels, Because of the reduced data rate. Offered by the compression techniques, computer network and Internet usage is becoming more and more image and graphic friendly, rather than being just data and text-centric phenomena. In short, high-performance compression has created new opportunities of creative applications such as digital library, digital archiving, video teleconferencing, telemedicine and digital entertainment to name a few. There are many other secondary advantages in data compression. For Example it has great implications in database access. Data compression may enhance the database performance because more compressed records can be packed in a given buffer space in a traditional computer implementation. This potentially increases the probability that a record being searched will be found in the main memory. Data security can also be greatly enhanced by encrypting the decoding parameters and transmitting them separately from the compressed database files to restrict access of proprietary information. An extra level of security can be achieved by making the compression and decompression processes totally transparent to unauthorized users. The rate of input-output operations in a computing device can be greatly increased due to shorter representation of data. Data compression obviously reduces the cost of backup and recovery of data in computer systems by storing the backup of large database files in compressed form. The advantages of data compression will enable more multimedia applications with reduced cost.

Data compression has some disadvantages too, depending on the application area and sensitivity of the data. For example, the extra overhead incurred by encoding and decoding processes is one of the most serious drawbacks of data compression, which discourages its usage in some areas. This extra overhead is usually required in order to uniquely identify or interpret the compressed data. Data compression generally reduces the reliability of the data records. For example, a single bit error in compressed code will cause the decoder to misinterpret all subsequent bits, producing incorrect data. Transmission of very sensitive compressed data (e.g., medical information) through a noisy communication channel (such as wireless media) is risky because the burst errors introduced by the noisy channel can destroy the transmitted data. Another problem of data compression is the disruption of data properties, since the compressed data is different from the original data.

In many hardware and systems implementations, the extra complexity added by data compression can increase the system‘s cost and reduce the system‘s efficiency, especially in the areas of applications that require very low-power VLSI implementation.

3. Image Compression Based on Entropy

The principle of digital image compression based on ‘information theory’. Image compression uses the concept of ‘Entropy’ to measure the amount of information that a source produces. The amount of information produced by a source is defined as its entropy. For each symbol, there is a product of the symbol probability and its logarithm. The entropy is a negative summation of the products of all the symbols in a given symbol set.

Compression algorithms are methods that reduce the number of symbols used to represent source information, therefore reducing the amount of space needed to store the source information or the amount of time necessary to transmit it for a given channel capacity. The mapping from source symbols into fewer target symbols is referred to as ‘compression’. The transformation from the ‘target symbols’ back into the source symbols representing a close approximation form of the original information is called ‘decompression’. Compression system consists of two steps, sampling and quantization of a signal. The choice of compression algorithm involves several conflicting considerations. These include degree of compression required, and the speed of operation. Obviously if one is attempting to run programs direct from their compressed state, decompression speed is paramount. The other consideration is size of compressed file versus quality of decompressed image. Compression is also known as encoding process and decompression is known as decoding process. Digital data compression algorithms can be classified into two categories:

- Lossless compression
- Lossy compression

Lossless compression:

In lossless image compression algorithm, the original data can be recovered exactly from the compressed data. It is used generally for discrete data such as text, computer generated data, and certain kinds of image and video information. Lossless

compression can achieve only a modest amount of compression of the data and hence it is not useful for sufficiently high compression ratios. GIF, Zip file format, and Tiff image format are popular examples of a lossless compression [30, 3]. Huffman Encoding and LZW are two examples of lossless compression algorithms. There are times when such methods of compression are unnecessarily exact.

In other words, 'Lossless' compression works by reducing the redundancy in the data. The decompressed data is an exact copy of the original, with no loss of data.

Lossy compression:

Lossy compression techniques refer to the loss of information when data is compressed. As a result of this distortion, must higher compression ratios are possible as compared to the lossless compression in reconstruction of the image. 'Lossy' compression sacrifices exact reproduction of data for better compression. It both removes redundancy and creates an approximation of the original.

The JPEG standard is currently the most popular method of lossy compression. The degree of closeness is measured by distortion that can be defined by the amount of information lost. Some example of lossless compression techniques are: ‘CCITT T.6’, Zip file format, and Tiff image format. JPEG Baseline and JPEG 2000 is an example of lossy compression algorithm. The three main criteria in the design of a lossy image compression algorithm are desired bit rate or compression ratio, acceptable distortion, and restriction on coding and decoding time. While different algorithms produce different type of distortion, the acceptability of which is often application dependent, there is clearly an increase in distortion with decreasing bit rate.

Obviously, a lossy compression is really only suitable for graphics or sound data, where an exact reproduction is not necessary. Lossy compression techniques are more suitable for images, as much of the detail in an image can be discarded without greatly changing the appearance of the image. In practice, very fine details are lost in image compression.

Redundancy different amount of data might be used. If the same information can be represented using different amounts of data, and the representations that require more data than actual information, is referred as data redundancy. In other

words, Number of bits required to represent the information in an image can be minimized by removing the redundancy present in it. Data redundancy is of central issue in digital image compression.

There are three kinds of redundancies that may present in the image and video.

- Coding redundancy
- lnterpixel redundancy
- Psychovisual redundancy

4. Transform coding

In transform coding, an image is transformed from one domain (usually spatial or temporal) to a different type of representation, using some well –known transform. Then the transformed values are coded and thus provide greater data compression. In this thesis, transforms are orthogonal so that the mapping is unique and reversible. As a result, the energy is preserved in the transform domain that is the sum of the squares of the transformed sequence is the same as the sum of the squares of the original sequence. Thus, the image can be completely recovered by the inverse transform.

Transform coding (TC), is an efficient coding scheme based on utilization of interpixel correlation. Transform coding uses frequency domain, in which the encoding system initially converting the pixels in space domain into frequency domain via transformation function. Thus producing a set of spectral coefficients, which are then suitably coded and transmitted.

The transform operation itself does not achieve any compression. It aims at decorrelating the original data and compacting a large fraction of the signal energy into a relatively small set of transform coefficients (energy packing property). In this way, many coefficients can be discarded after quantization and prior to encoding. Most practical transform coding systems are based on DCT of types II which provides good compromise between energy packing ability and computational complexity. The energy packing property of DCT is superior to that of any other unitary transform [45]. Transforms that redistribute or pack the most information into the fewest coefficients provide the best sub-image approximations and, consequently, the smallest reconstruction errors. In Transform coding, the main idea is that if the transformed version of a signal is less correlated compared with the original signal, then quantizing and encoding the transformed signal may lead to data compression. At the receiver, the encoded data are decoded and transformed back to reconstruct the signal.

The purpose of the transform is to remove interpixel redundancy (or de-correlate) from the original image representation. The image data is transformed to a new representation where average values of transformed data are smaller than the original form. This way the compression is achieved. The higher the correlation

among the image pixels, the better is the compression ratio achieved. There are various methods of transformations being used for data compression as follows:

- Karhunen-Loeve Transform (KLT)
- Discrete Fourier Transform (DFT)
- Discrete Sine Transform (DST)
- Walsh Hadamard Transform (WHT)
- Discrete Cosine Transform (DCT)
- Discrete Wavelet Transform (DWT)

Transform-based Image Compression:

Transform refers to changing the coordinate basis of the original signal, such that a new signal has the whole information in few transformed coefficients. The processing of the signals in the transform domain is more efficient as the transformed coefficients are not correlated.

A popular image compression framework is the transform based image compression as shown in below Figure 2.3

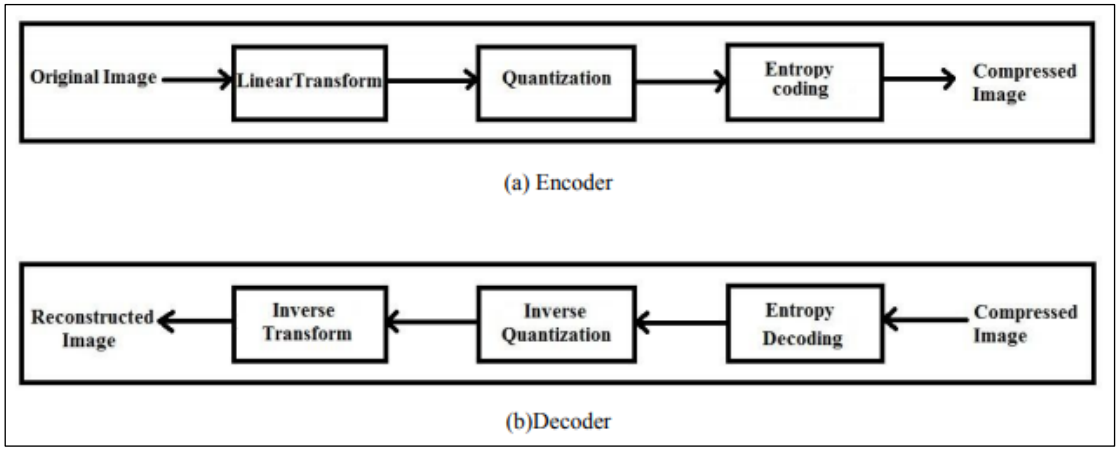


Figure 4.1 Block diagram of transform based image coder (a) Compression or Encoder (b) Decompression or Decoder

The first step in the encoder is to apply a linear transform to remove redundancy in the data, followed by quantizing the transform coefficients, and finally entropy coding then we get the quantized outputs. After the encoded input image is transmitted over the channel, the decoder reverse all the operations that are applied in the encoder side and tries to reconstruct a decoded image as close as to the original image.

5. Discrete Cosine Transform:

DCT is an orthogonal transform, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency. The DCT transforms a signal from a spatial representation into a frequency representation. The DCT represent an image as a sum of sinusoids of varying magnitudes and frequencies. DCT has the property that, for a typical image most of the visually significant information about an image is concentrated in just few coefficients of DCT. After the computation of DCT coefficients, they are normalized according to a quantization table with different scales provided by the JPEG standard computed by psycho visual evidence. Selection of quantization table affects the entropy and compression ratio. The value of quantization is inversely proportional to quality of reconstructed image, better mean square error and better compression ratio. In a lossy compression technique, during a step called Quantization, the less important frequencies are discarded, and then the most important frequencies that remain are used to retrieve the image in decomposition process. After quantization, quantized coefficients are rearranged in a zigzag order for further compressed by an efficient lossy coding algorithm. DCT has many advantages:

6. Conclusion

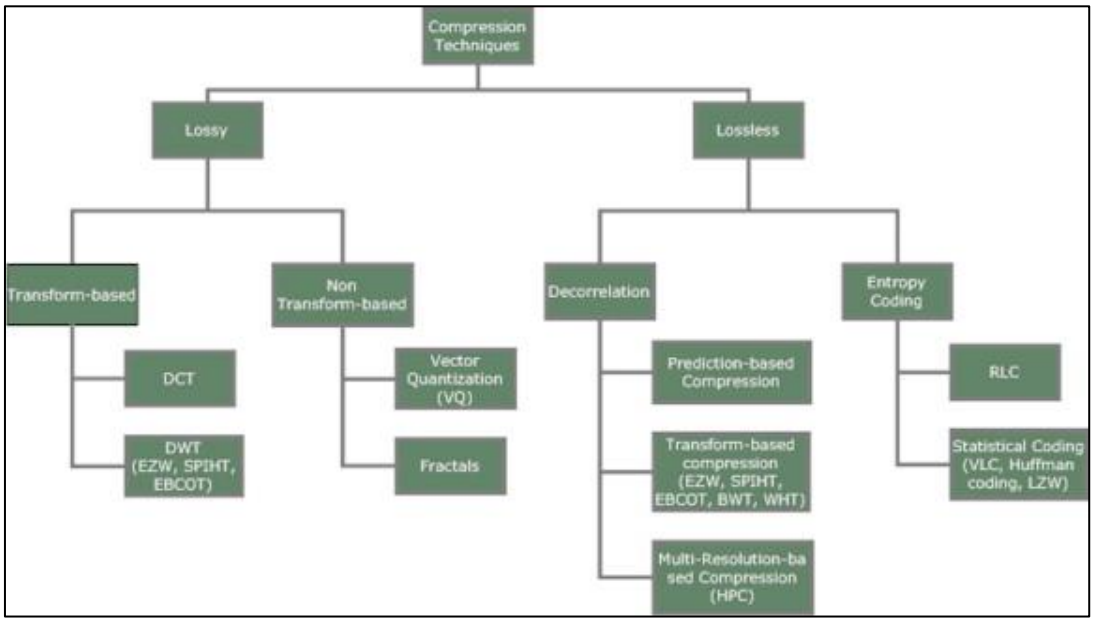


Figure 6.1: Compression Techniques

The above figure depict the entire discussion, and I can conclude that My Project "Image Compressor" uses transform based lossy compression technique known as DCT (Discrete Cosine Transformation).

Definition of Problem

The increasing demand for multimedia content such as digital images has led to great interest in research into compression tools. The development of higher quality and less expensive image acquisition devices has produced steady increases in both image size and resolution, and a greater consequent for the design of efficient compression tools. Although storage capacity and transfer bandwidth has grown accordingly in recent years, many applications still require compression.

In general, this project investigates all image file formats and the compression tools available and building a simple image compressing software. The main objective is to design and develop compressing software for windows operating system to make efficient use of storage devises and bandwidth available in the transmission medium. Factors related to the need for image compression include:

- The large storage requirements for multimedia data
- Effective use of the transmission medium
- Low power devices such as handheld phones have small storage capacity
- Network bandwidths currently available for

Image compression plays very important role in storing and transferring images efficiently. With the evolution of electronic devices like Digital Cameras, Smart phones, biometric applications, the requirement of storing images in less memory is becoming necessary. Using image compressing software the requirement of less memory storage and efficient transmission can be fulfilled. At the same time it should be considered that, the compressing software must be able to compress the image with efficient loss as per the user requirement. The objective of the project is to have such compressing software which will allow user to select the level of compression he or she wishes to perform on the image.

Hence, I can conclude, the main problem identified to build this project is to facilitate the following:

- Store data in an efficient form
- Transmit data in an efficient form

The actual solution to these problems is to have an image compressing mechanism which will allow user to be able to compress his or her images.

A data compression system mainly consists of three major steps and that are removal or reduction in data redundancy, reduction in entropy, and entropy encoding. A typical data compression system can be labelled using the block diagram shown in Figure 2.1 It is performed in steps such as image transformation, quantization and entropy coding. JPEG is one of the most used image compression standard which uses discrete cosine transform (DCT) to transform the image from spatial to frequency domain. An image contains low visual information in its high frequencies for which heavy quantization can be done in order to reduce the size in the transformed representation. Entropy coding follows to further reduce the redundancy in the transformed and quantized image data.

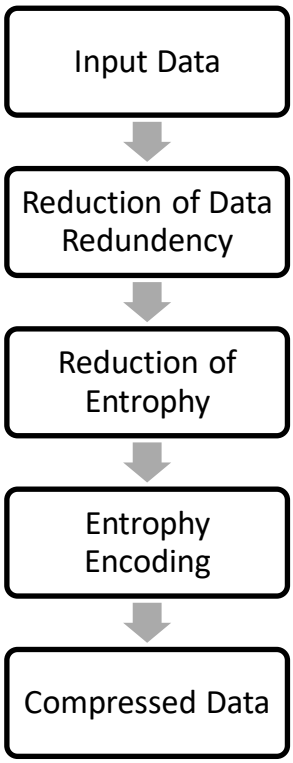


Figure: Data Compression Model

System analysis and design and user's requirements

1. Feasibility Study

A Feasibility Study determines whether a project is worth doing. The process followed for making this determination is called "Feasibility Study". This type of study determines whether a project can and should proceed.

The following are the various types of feasibility studies that are undertaken in this project.

Technical Feasibility:

- As it is a windows application it is technically feasible, just a Laptop or PC with required software such as Visual Studio will be enough to make the software.
- Besides this, development requires skilled people who are able to develop a windows based application using Visual Studio.

Economic Feasibility:

- Since most of the software tools can be freely found on the Microsoft Platform hence, it is economically feasible.
- Also, the software can be found with open source licence so, there is no issue after the completion.
- Also, its feasibility can be found to depend on the use of software used since all the software can be found with cost and without cost.

Legal Feasibility:

- It will be 100% legally feasible as the data is not shared between any type of entity because, it is just a standalone software which will not needs access to the user information or internet.

Scheduling Feasibility:

- Feasibility depends on the person skills in windows application development.

Behavioural Feasibility:

- The application requires no special technical guidance and all the users since it depends on the user's knowledge of using commuter applications.
- Since there is a web platform "www.4imagecompressor.blogspot.in" for the users to guide them with warning and failure messages for all the actions taken hence, the project is behaviourally feasible.

Operational feasibility:

- Since the application is able to perform well as per the requirements, it can work with any type of image file format, any number of image file, hence it is operationally feasible.
- Also application is able to handle almost all type of errors and exceptions as a result, success of operational fusibility can be considered.

2. Tools and Technologies

Tools:

- Visual Studio 2015 Community:

  A rich, integrated development environment for creating stunning applications for Windows, Android, and iOS, as well as modern web applications and cloud services. Free, full-featured and extensible tool for developers building non-enterprise applications.
- Confuser:

  A powerful and customizable obfuscator for .NET applications that let you select algorithms by which to secure source code, making it difficult to reverse-engineer.
- .Net Reflector:

  This is a versatile and intuitive application that helps you to decompile, understand, and fix any .NET code, even if you don't have the source.
- Inno Setup:

Tool that helps you to create Windows installers in the form of single EXE files to install your program for easy online distribution.

Technologies:

- .NET framework:

  .NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (in contrast to a hardware environment) named Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. (As such, computer code written using .NET Framework is called "managed code".) FCL and CLR together constitute .NET Framework.

  List of .Net Framework versions are as follows:

  - .NET Framework 1.0
  - .NET Framework 1.1
  - .NET Framework 2.0
  - .NET Framework 3.0
  - .NET Framework 3.5
  - .NET Framework 4
  - .NET Framework 4.5
  - .NET Framework 4.5.1
  - .NET Framework 4.5.2
  - .NET Framework 4.6
  - .NET Framework 4.6.1
  - .NET Framework 4.6.2
  - .NET Framework 4.7
  - .NET Framework 4.7.1

- C#.Net Programming:

C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure.

- ISS Scripting:

  ISS files store all of the information that is used to compile a program installer in the Inno Setup Compiler. By default, compiled programs are generated to an "Output" directory within the ISS script directory. ISS files can be edited using the Inno Setup Compiler or any text editor. They are similar to .INI files.

3. User Requirements

"Image Compressor" will be able to work in a computer or laptop with the following hardware and software components.

- Hardware Components
  - Any available CPU
  - At least 100MB of free hard disk of any available size
  - Any available size of RAM
  - Any available resolution of display
- Software Components
  - Windows 7 / 8 / 8.1 / 10
  - .NET Framework 4.0 or above

System Planning

1. PERT chart

A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project linked by labelled vectors (directional lines) representing tasks in the project. The direction of the arrows on the lines indicates the sequence of tasks.

The following table shows the planning of my project as Task Name, Duration, Start Date, Finish Date, Predecessor, Resource Name etc.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Defining Problem	10 days	Fri 01-12-17	Thu 14-12-17		Noting
2		Defining System	10 days	Fri 15-12-17	Thu 28-12-17	1	Nothing
3		Project Schedule	5 days	Fri 29-12-17	Thu 04-01-18	2	MS Project 2010
4		Design UI	25 days	Fri 05-01-18	Thu 08-02-18	3	Visual Studio 2015 Community
5		Writing Code	30 days	Fri 09-02-18	Thu 22-03-18	4	Visual Studio 2015 Community
6		Unit Testing	5 days	Fri 23-03-18	Thu 29-03-18	5	Visual Studio 2015 Community
7		Integration Testing	5 days	Fri 30-03-18	Thu 05-04-18	6	Visual Studio 2015 Community
8		System Testing	5 days	Fri 06-04-18	Thu 12-04-18	7	Visual Studio 2015 Community
9		Deplyment	5 days	Fri 13-04-18	Thu 19-04-18	8	www.4imagecompressor.blogspot.in

Figure 1.1: System Planning

The PERT Chart for the above system planning can be given as follows:

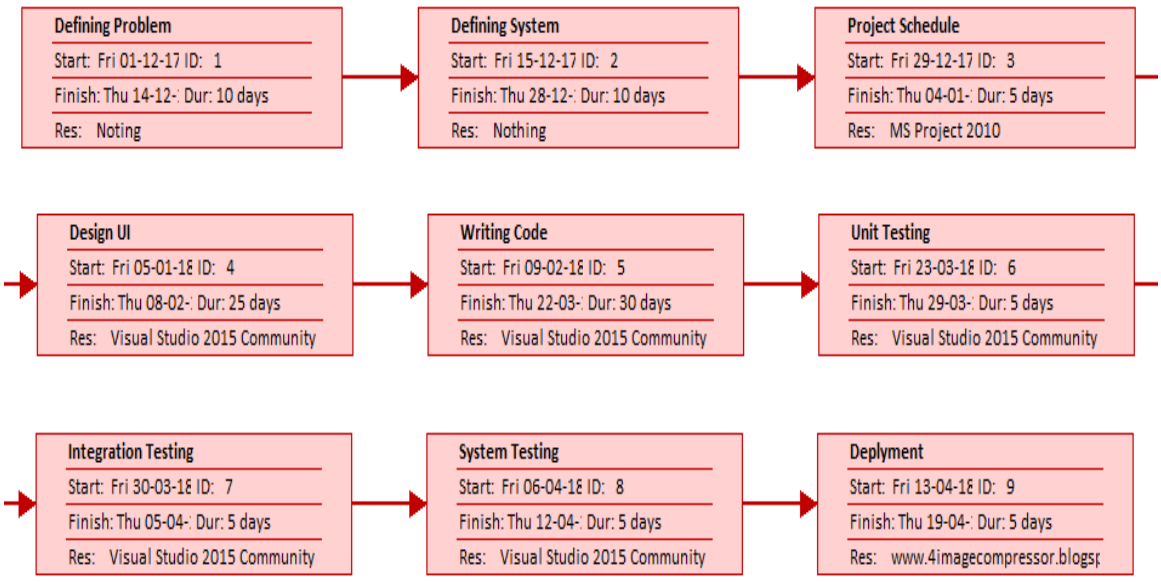


Figure 1.2: PERT chart

2. Gantt chart

The PERT chart is sometimes preferred over the Gantt chart, another popular project management charting method, because it clearly illustrates task dependencies. On the other hand, the PERT chart can be much more difficult to interpret, especially on complex projects. Frequently, project managers use both techniques.

The Gantt chart for the above mentioned system planning can be given as follows:

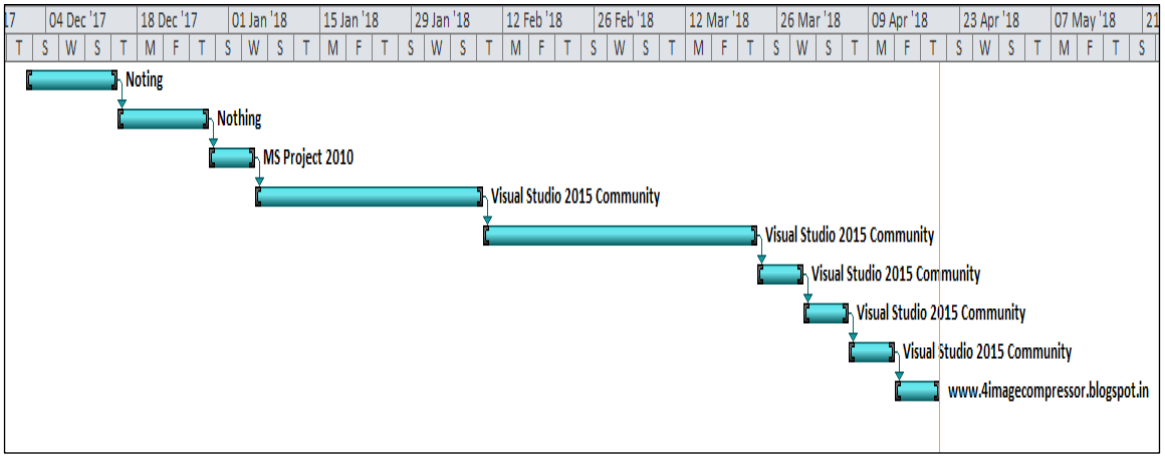


Figure 2.1: Gantt chart

Detailed life cycle of the project

UML is the best modelling language through which I can surely be able to explain the complete detailed life cycle of my project “Image Compressor”. The Unified Modelling Language (UML) is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

UML Model for Compression of Image: Step involved in compressing an image is as follows.

- Load image
- Take a sub image of size 8x8
- Apply cosine transform for this sub image
- Then these coefficients are normalized using the normalized matrix and apply run length coding technique to compress the image.
- Repeat the same procedure until all image are scanned.
- Exit

Three main aspects of UML i.e. Class Diagram, Activity Diagram and Sequence Diagram can help anyone to be able to understand the project and build accordingly.

- UML Class Diagram, which shows how the classes are associated to each other for compressing any image.
- UML Activity Diagram, which shows systematic steps involved in compressing any image of any type.
- UML Sequence Diagram, one can see that how message passing take place among the different classes like Load_image, Subimage, DCT_transform, Quantization, Run_length_code and Compress_image. It also shows that how long time that all classes involve in Execution.

1. UML Class Diagram

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

In the diagram, classes are represented with boxes which contain three parts:

- The upper part holds the name of the class.
- The middle part contains the attributes of the class.
- The bottom part gives the methods or operations the class can take or undertake

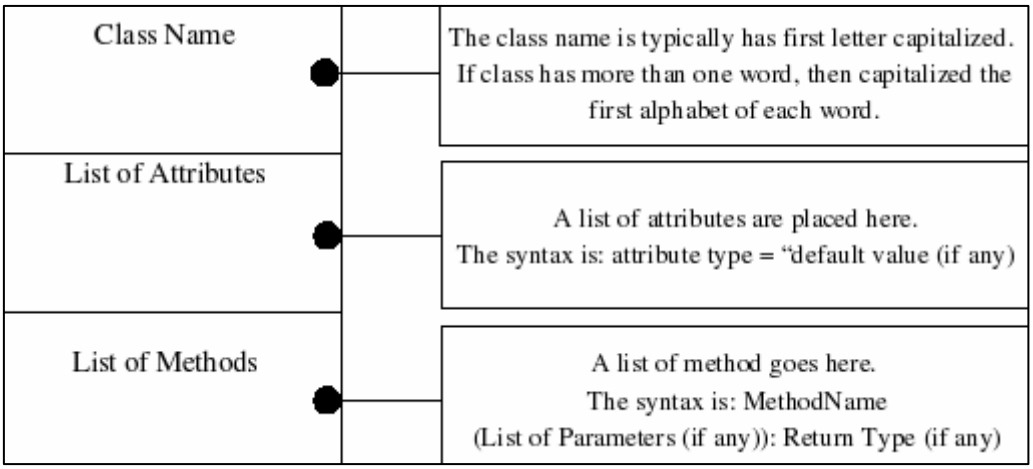


Figure 1.1: Class Diagram Elements

In the design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

Multiplicity:

- 0..1 no instance or one instance
- 1 Exactly one instance
- 0..* Zero or more instance
- 1..* One or more instance

There may various relations can exist between the classes these rotations can be shows by one of the following link depending on the type of relation.

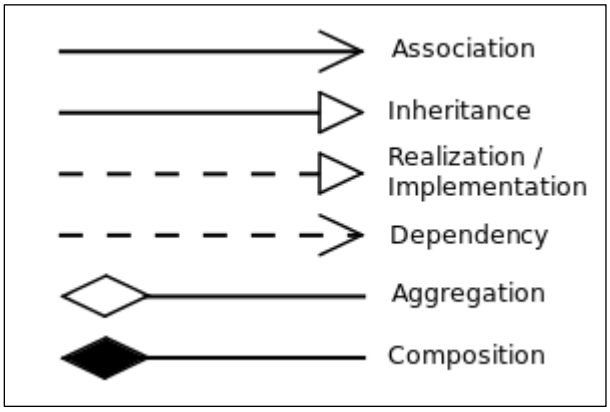


Figure 1.2: Different Links in Class Diagram

Hence, the UML Class Diagram for my project “Image Compressor” will be as follows:

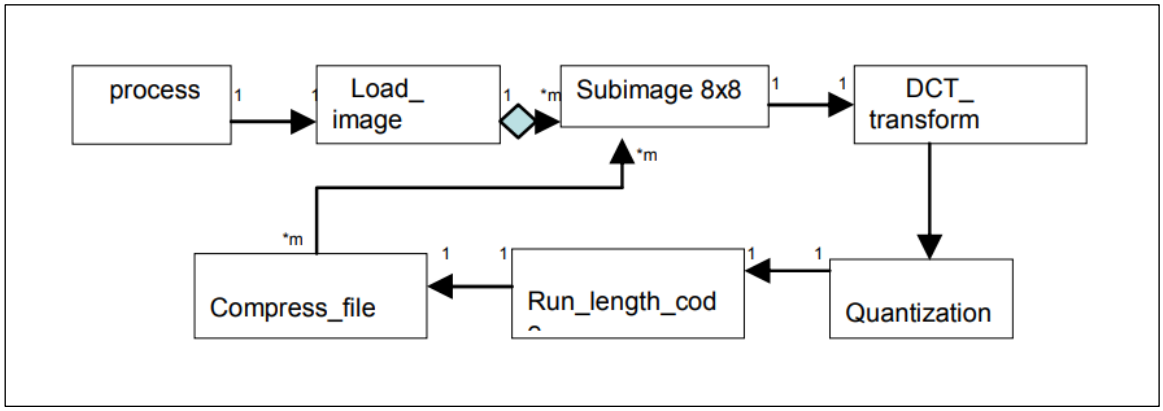


Figure 1.3: UML Class Diagram for Image Compressor

2. UML Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- Rounded rectangles represent actions
- Diamonds represent decisions
- Bars represent the start (split) or end (join) of concurrent activities
- Black circle represents the start (initial state) of the workflow
- Encircled black circle represents the end (final state).

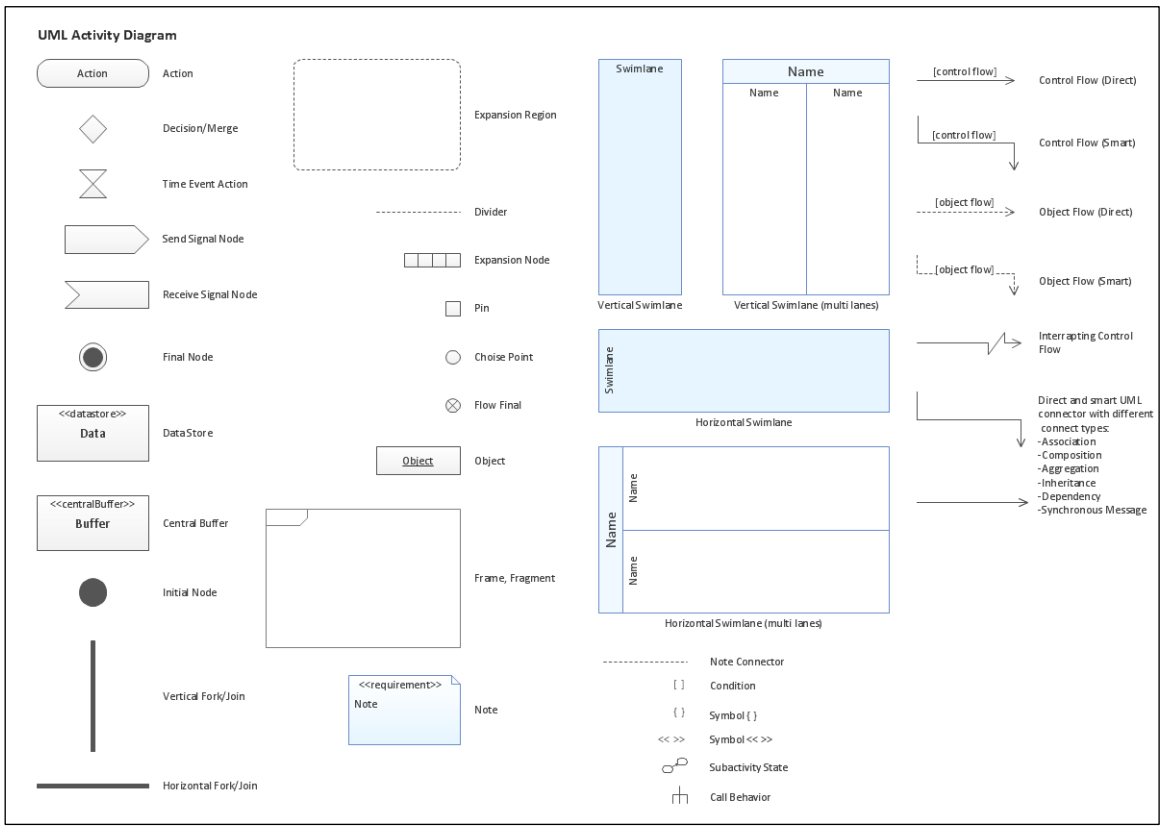


Figure 2.1: Activity Diagram Elements

Arrows run from the start towards the end and represent the order in which activities happen. Hence, they can be regarded as a form of flowchart.

As a result, the UML Activity Diagram for my project “Image Compressor” will be as follows:

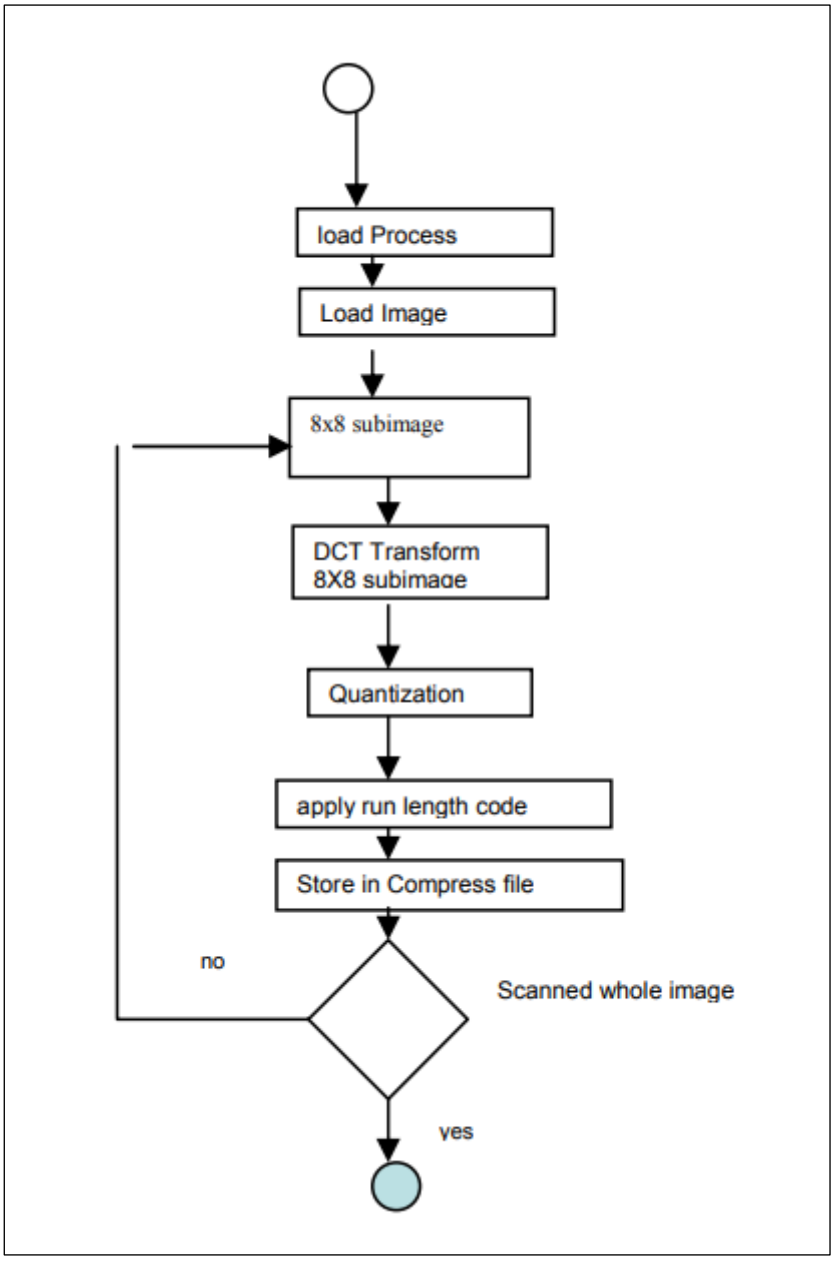


Figure 2.2: UML Activity Diagram for Image Compression

3. UML Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Some of the elements of the sequence diagram are as follows:

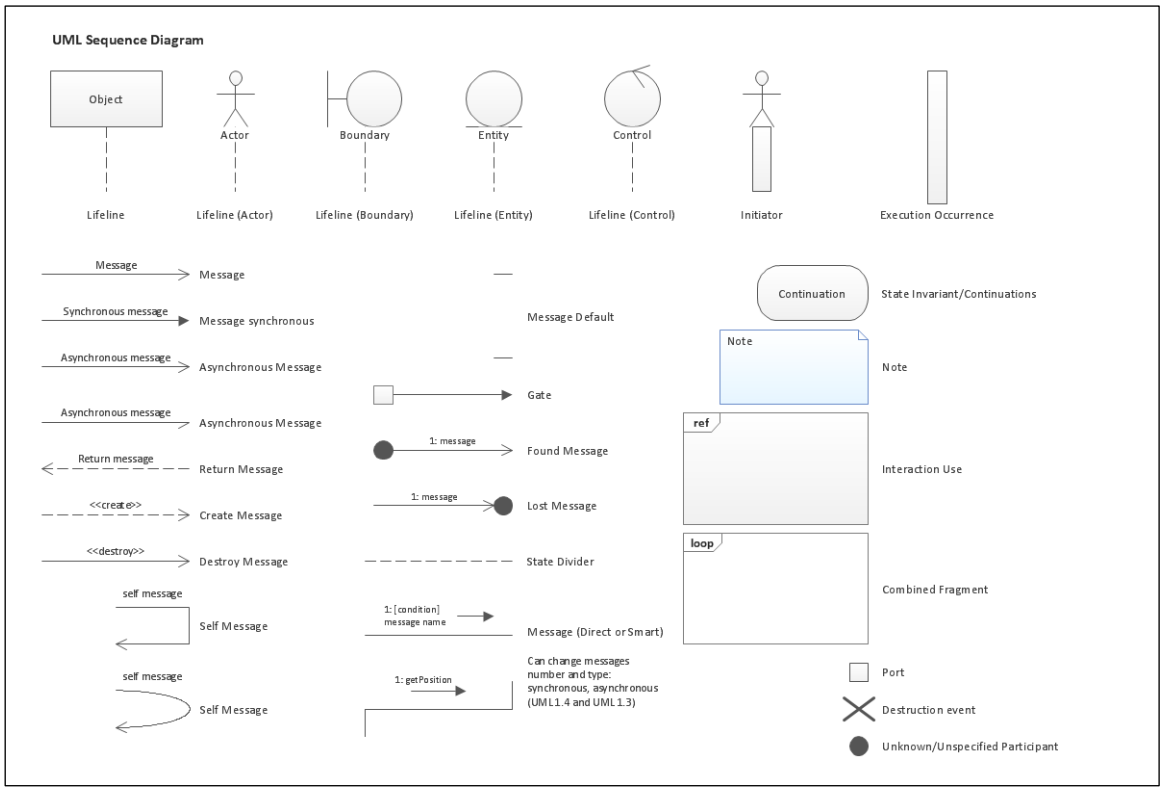


Figure 3.1: Sequence Diagram Elements

Hence, the UML Sequence Diagram for my project “Image Compressor” will be as follows:

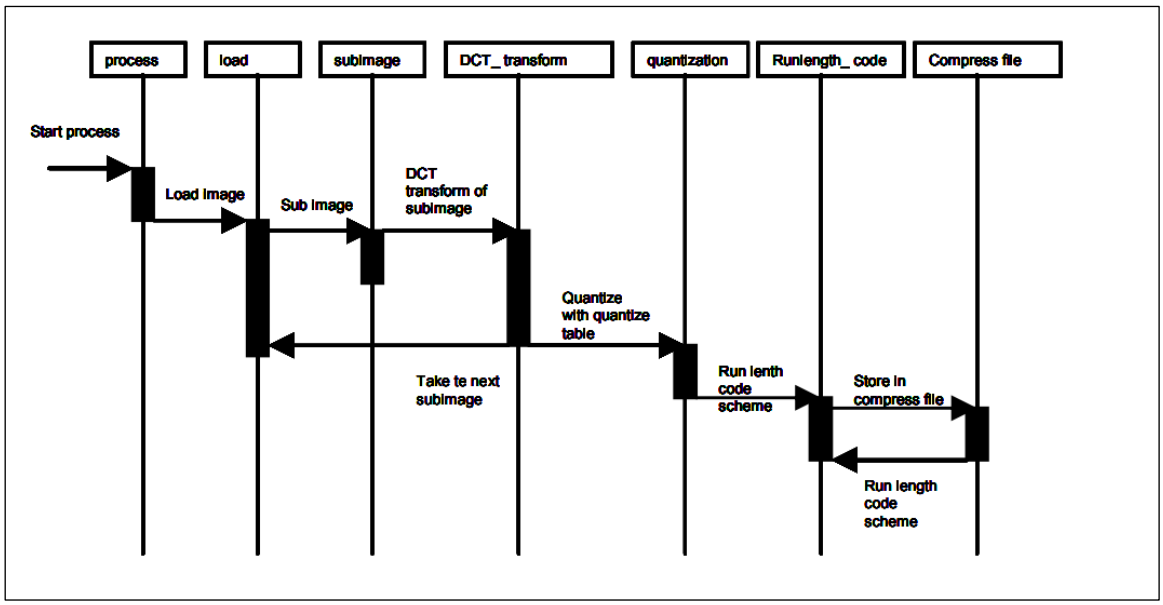


Figure 3.2: UML Sequence Diagram for Image Compression

Process involved

1. Process model

A Process model describes the sequence of phases for the entire lifetime of a product. Therefore, it is sometimes also called Product Life Cycle. This covers everything from the initial commercial idea until the final de-installation or disassembling of the product after its use. Process Model provides a fixed framework that guides a project in:

- Development of the product
- Planning and organizing the project
- Tracking and running the project

There are many Process Models to achieve this they are as follows:

- Waterfall Model
- The V-Model
- Incremental Model
- Agile Model
- Iterative Model
- Spiral Model
- Prototype Model
- Code-and-Fix Model
- Extreme Programming (XP) Model
- Rapid Application Development (RAD) model

By analyzing the complete life cycle of my project I can conclude that i have used "Incremental Model" of developing a complete project. Some of the details about the "Incremental Model" of development are as follows:

2. Incremental model:

In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a "multi-waterfall" cycle. Cycles are divided up into smaller, more easily managed modules. Incremental model is a type of software development model like V-model, Agile model etc.

In this model, each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

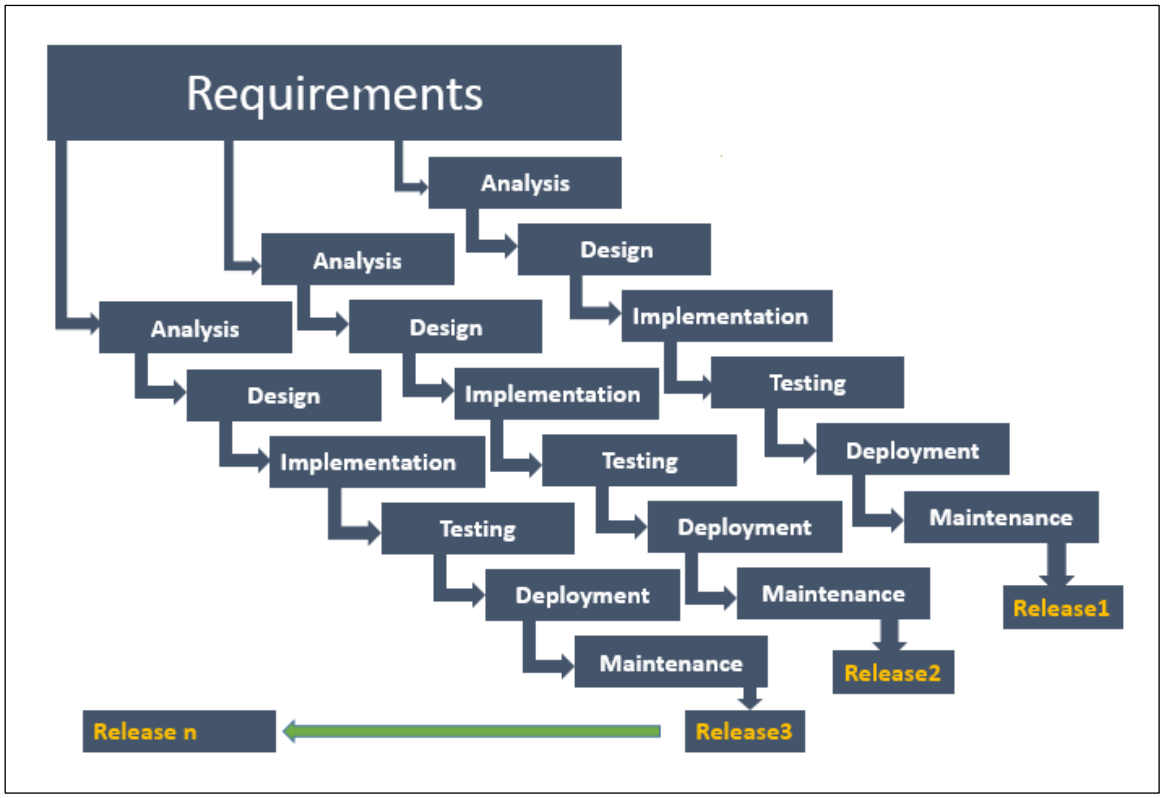


Figure 2.1: Incremental Model

Some of the advantages of incremental model are as follows:

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

Some of the disadvantages of incremental model are as follows:

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Total cost is higher than waterfall.

Some reasons why i chosen this model include the following:

- This model can be used when the requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some details can evolve with time.
- There is a need to get a product to the market early.
- A new technology is being used
- Resources with needed skill set are not available
- There are some high risk features and goals.

Methodology adopted, system implementation and details of hardware and software used

1. Methodology adopted:

The complete project “Image Compressor” is built based on object oriented methodology. Since, C#.Net is a modern programming language which support object oriented methodology of development.

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that interact with one another. There is significant diversity of OOP languages, but the most popular ones are class-based, meaning that objects are instances of classes, which typically also determine their type.

Many of the most widely used programming languages (such as C++, Object Pascal, Java, Python etc.) are multi-paradigm programming languages that support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming. Significant object-oriented languages include Java, C++, C#, Python, PHP, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Common Lisp, and Smalltalk.

The conceptual framework of object–oriented systems is based upon the object model. There are two categories of elements in an object-oriented system:

Major Elements: By major, it is meant that if a model does not have any one of these elements, it ceases to be object oriented. The four major elements are: Abstraction, Encapsulation, Modularity and Hierarchy.

Minor Elements: By minor, it is meant that these elements are useful, but not indispensable part of the object model. The three minor elements are: Typing, Concurrency and Persistence.

2. System implementation:

.Net framework and C#.Net programming has huge set of APIs, working with them one can easily implement the Image Compressing mechanism. Some of the aspect that I have used is as follows:

Using Image Encoders and Decoders in Managed GDI+:

The System.Drawing namespace provides the Image and Bitmap classes for storing and manipulating images. By using image encoders in GDI+, you can write images from memory to disk. By using image decoders in GDI+, you can load images from disk into memory. An encoder translates the data in an Image or Bitmap object into a designated disk file format. A decoder translates the data in a disk file to the format required by the Image and Bitmap objects.

GDI+ has built-in encoders and decoders that support the following file types:

- BMP
- GIF
- JPEG
- PNG
- TIFF

GDI+ also has built-in decoders that support the following file types:

- WMF
- EMF
- ICON

Using Transformations in Managed GDI+:

Affine transformations include rotating, scaling, reflecting, shearing, and translating. In GDI+, the Matrix class provides the foundation for performing affine transformations on vector drawings, images, and text.

How to: Set JPEG Compression Level:

You may want to modify the parameters of an image when you save the image to disk to minimize the file size or improve its quality. You can

adjust the quality of a JPEG image by modifying its compression level. To specify the compression level when you save a JPEG image, you must create an EncoderParameters object and pass it to the Save method of the Image class. Initialize the EncoderParameters object so that it has an array that consists of one EncoderParameter. When you create the EncoderParameter, specify the Quality encoder, and the desired compression level.

Example: The following example code creates an EncoderParameter object and saves three JPEG images. Each JPEG image is saved with a different quality level, by modifying the long value passed to the EncoderParameter constructor. A quality level of 0 corresponds to the greatest compression, and a quality level of 100 corresponds to the least compression.

```
private void VaryQualityLevel()
    {
        // Get a bitmap. The using statement ensures objects
        // are automatically disposed from memory after use.
        using (Bitmap bmp1 = new Bitmap(@"C:\TestPhoto.jpg"))
        {
            ImageCodecInfo jpgEncoder = GetEncoder(ImageFormat.Jpeg);

            // Create an Encoder object based on the GUID
            // for the Quality parameter category.
            System.Drawing.Imaging.Encoder myEncoder =
                System.Drawing.Imaging.Encoder.Quality;

            // Create an EncoderParameters object.
            // An EncoderParameters object has an array of EncoderParameter
            // objects. In this case, there is only one
            // EncoderParameter object in the array.
            EncoderParameters myEncoderParameters = new EncoderParameters(1);

            EncoderParameter myEncoderParameter = new EncoderParameter(myEncoder, 50L);
            myEncoderParameters.Param[0] = myEncoderParameter;
            bmp1.Save(@"c:\TestPhotoQualityFifty.jpg", jpgEncoder, myEncoderParameters);

            myEncoderParameter = new EncoderParameter(myEncoder, 100L);
            myEncoderParameters.Param[0] = myEncoderParameter;
            bmp1.Save(@"C:\TestPhotoQualityHundred.jpg", jpgEncoder, myEncoderParameters);

            // Save the bitmap as a JPG file with zero quality level compression.
            myEncoderParameter = new EncoderParameter(myEncoder, 0L);
            myEncoderParameters.Param[0] = myEncoderParameter;
            bmp1.Save(@"C:\TestPhotoQualityZero.jpg", jpgEncoder, myEncoderParameters);
        }
    }
```

Figure 10.2: Set JPEG Compression Level

3. Details of hardware and software used:

Hardware used:

- Toshiba Satellite C50-A I2011
  - Screen Size: 15.6 inches
  - Processor: Intel Core i3 2348M
  - RAM: 4 GB
  - Hard Drive: 640 GB

Software used:

- Windows 8.1 Pro 64-bit
  - Visual Studio 2015 Community: A rich, integrated development environment for creating stunning applications for Windows, Android, and iOS, as well as modern web applications and cloud services. Free, full-featured and extensible tool for developers building non-enterprise applications.
  - Confuser: A powerful and customizable obfuscator for .NET applications that let you select algorithms by which to secure source code, making it difficult to reverse-engineer.
  - .Net Reflector: This is a versatile and intuitive application that helps you to decompile, understand, and fix any .NET code, even if you don't have the source.
  - Inno Setup: Tool that helps you to create Windows installers in the form of single EXE files to install your program for easy online distribution.

System maintenance and evaluation

1. System maintenance

Since the software is robust and will work properly and efficiently. There are certain steps or conditions that the user must look into, so the lifetime of the software can be increased and thus already maintained. When system is allowed to run for sometimes they tend to become disorganized.

This results in system inefficiently. To provide a course of action, where by the inefficiency in the system is arrested and state of the system is brought back to equilibrium, we undertake certain measure. This is done with the help of maintenance of the system.

The maintenance of the system is very much essential in the long run of the system. This maintenance can only be carried out if a proper evaluation of the system is carried our if a proper evaluation of the system is carried out if a proper evaluation of the system is carried out if a proper maintenance of the system is carried out at regular intervals. So the schemes for maintenance of the system are to be pre-determined before the actual implementation of the system.

- Always work with support of higher authorities.
- Always keep other windows closed while on this software.
- Read the user manual carefully before using the software.
- A periodic review of the system at regular intervals.
- A meeting with the client to access the current utility of the system and

Some of those conditions are as follows:

- level of satisfaction.
- Bringing out system modification.

Problems in the system arise due to business environment changes, leading to change in demand and wants of the client. Such a change in the requirement factors call for modification of the system to revised information.

Broadly we can state the major problems can be as follows:

- Making the required modification to suit the needs of the client.
- As the business environment has changed the system needs to be changed as well

From the above discussion I can conclude that my software does not require a dedicated maintenance plan since, it's standalone software, if one copy of the software started working unusual the user can get and use the other copy of the software from www.4imagecompressor.blogspot.com

Hence maintenance of the deployment platform is mandatory to allow the user to be able to get copies of the software and have a effective guidance to use the software.

2. Limitations

The lists of limitations of this project can be quite big as a end user has many requirements to be fulfilled.

Considering the functionality provided till now the limitations are as follows:

- Icon image (.ico) files are not supported by the application.
- At once only one compression operation can be performed.
- Each time the compression operation is completed the user need to restart the application from scratch to perform other operation.
- No interactive functionality to inform the user for completion of the compression operation.

3. Future enhancements

Talking about future enhancement, then we have a huge list for it. These will be implemented as per the popularity and need among the people.

Some of them are as follows:

- Enabling user to be able to see the effect of level of compression before the actual compression.
- Connecting the software application through net so that user can get the notification for the new version of the application.

- Enabling user to be able to work with the application with their images on the cloud.
- Software should be able be log uncaught exception in a text file so that developer can have list of unhandled exceptions to develop further versions of the same application.
- Increase in performance by decreasing the compression time required.

Cost and benefit analysis

Cost–benefit analysis (CBA), sometimes called benefit costs analysis (BCA), is a systematic approach to estimate the strengths and weaknesses of alternatives (for example in transactions, activities, and functional business requirements or projects investments). It is used to determine options that provide the best approach to achieve benefits while preserving savings.

The CBA is also defined as a systematic process for calculating and comparing benefits and costs of a project. CBA has two main purposes:

- To determine if an investment/decision is sound (justification/feasibility) – verifying whether its benefits outweigh the costs, and by how much
- To provide a basis for comparing projects – which involves comparing the total expected cost of each option against its total expected benefits.

My project “Image Compressor” is a windows application which uses the following hardware and software to build it which was already available to me for free.

- Hardware:
  - Toshiba Satellite C50-A I2011
    - ✓ Screen Size: 15.6 inches
    - ✓ Processor: Intel Core i3 2348M
    - ✓ RAM: 4 GB
    - ✓ Hard Drive: 640 GB
- Software:
  - Windows 8.1 Pro 64-bit
    - ✓ Visual Studio 2015 Community
    - ✓ Confuser
    - ✓ .Net Reflector
    - ✓ Inno Setup

Also, due to the reach facility .NET framework, C#.Net language and Microsoft Developer Community, I was able to build “Image Compressor” with less LOCs.

Just the time and effort is the real cost of this project which gives a successful project completion as a benefit. This project is mainly made to be knows as a windows software developer and as an academic project. Hence, I can conclude that i got approximately 100% ROI on this project in two aspects.

- Completion of academic year project
- To be knows as developer of “Image Compressor” windows application

Methodology used for testing

Considering the properties each testing methodologies have, I can say I have used the following testing techniques for successful development of my project. These are as follows:

- Unit testing
- Integration testing
- System testing

1. Unit testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Advantages of Unit testing:

- Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.
- Reduces Cost of Testing as defects are captured in very early phase.
- Improves design and allows better refactoring of code.
- Unit Tests, when integrated with build gives the quality of the build as well.

Unit Testing Techniques:

- Black Box Testing - Using which the user interface, input and output are tested.
- White Box Testing - used to test each one of those functions behaviour is tested.
- Gray Box Testing - Used to execute tests, risks and assessment methods.

For the project “Image Compressor”, I used Visual Studio 2015 only to perform the unit testing without any test script, manually.

2. Integration testing

Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

Integration Strategies:

- Big-Bang Integration
- Top Down Integration
- Bottom Up Integration
- Hybrid Integration

Also, to perform the integration testing, I used Visual Studio 2015 Community, without any test script, manually.

3. System testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

Some of the types of system tests are:

- Functionality
- Inter-Operability
- Performance
- Scalability
- Stress
- Load and Stability
- Reliability
- Regression
- Regulatory and Compilation

For system testing, initially i used Visual studio 2015 Community without test script, manually, then to test non-functional attributes such as security, I used .Net Reflector. The software was able to reverse engineer “Image Compressor”. Confuser, a software helped to make “Image compressor” resist reverse engineering.

The following figures depict the process:

This is the figure shows if the software is tested for security i.e. for reverse engineering it can be done using .Net Reflector and the entire source code can be seen.

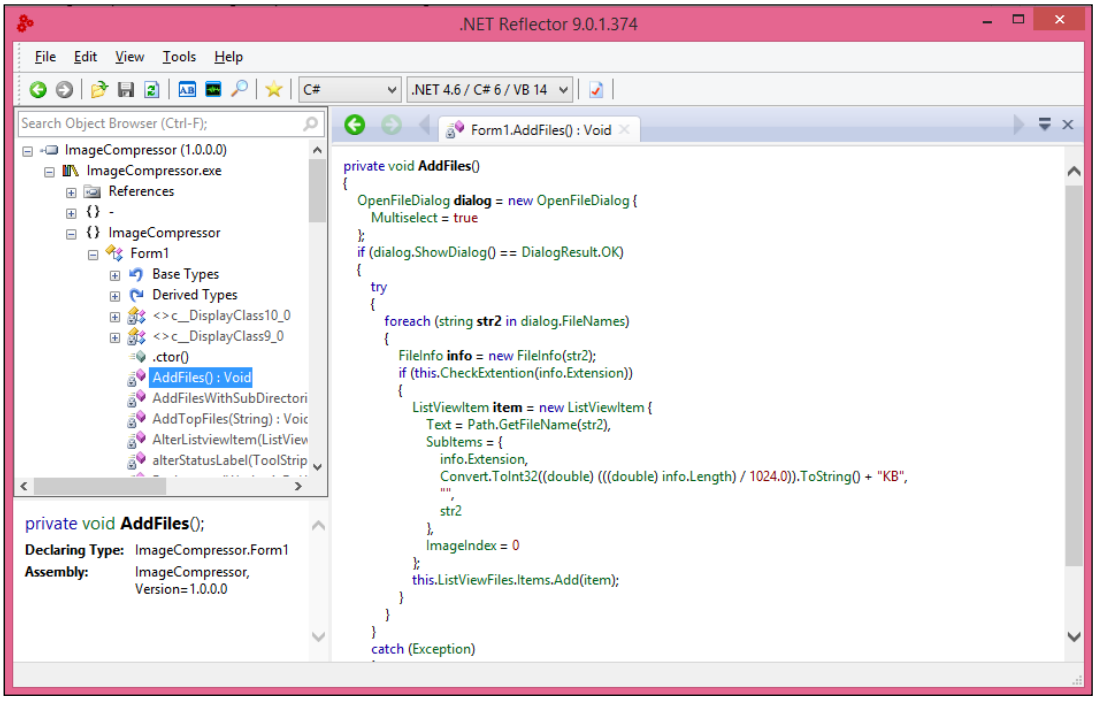


Figure 3.1: Image Compressor when reverse engineered

Now, to make application resist reverse engineering, I am using Confuser to make the software confused so that it can resist reverse engineering.

The following two figure depicts the successful process of confusing with the default configuration of confuser tool after which when we try to reverse engineer the software “Image Compressor” using .Net Reflector it resist and we actually can’t see the source code of the application.

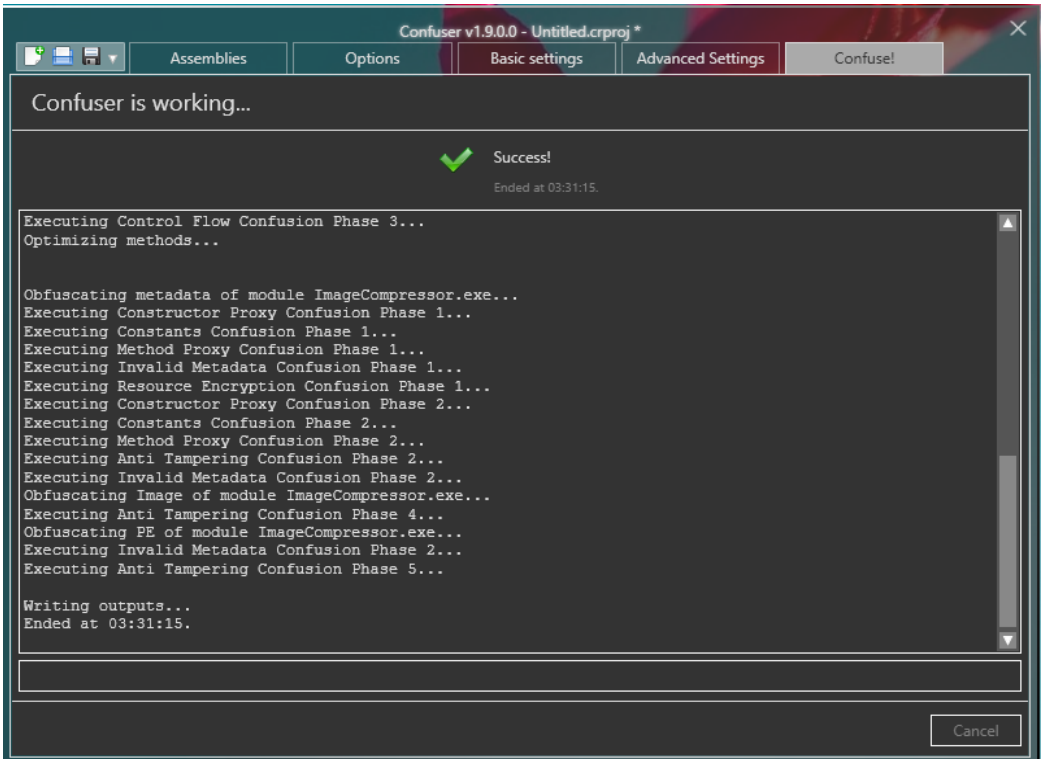


Figure 3.2: Confusing Image Compressor

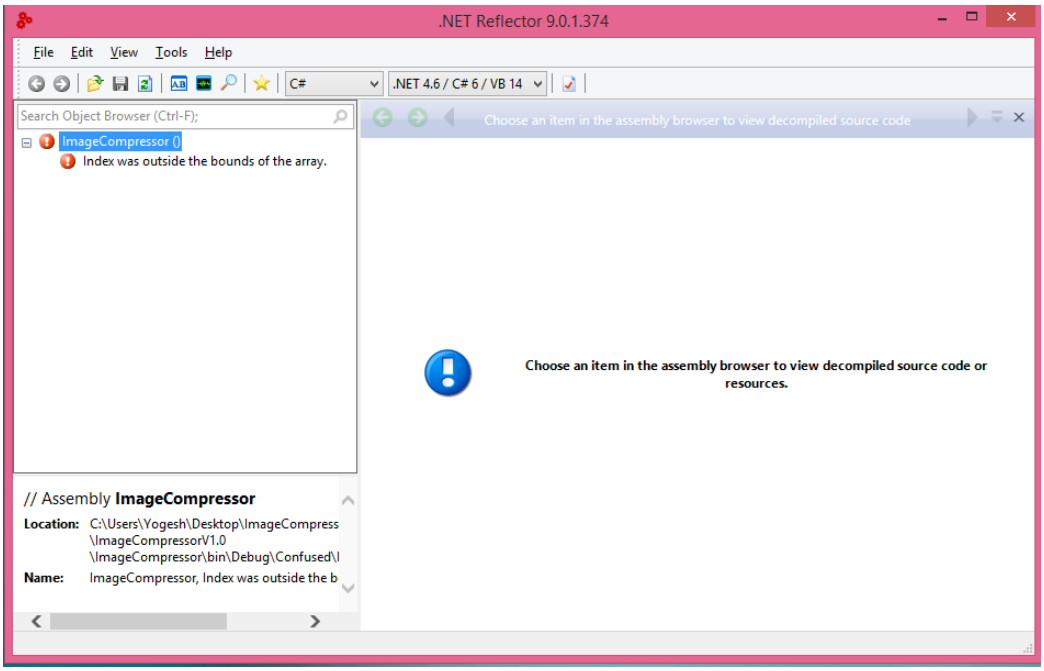


Figure 2.3: Reverse engineering confused Image Compressor

User or operational manual

The following steps guide you in downloading the “Image Compressor” software.

Step 1: Google www.4imagecompressor.blogspot.in

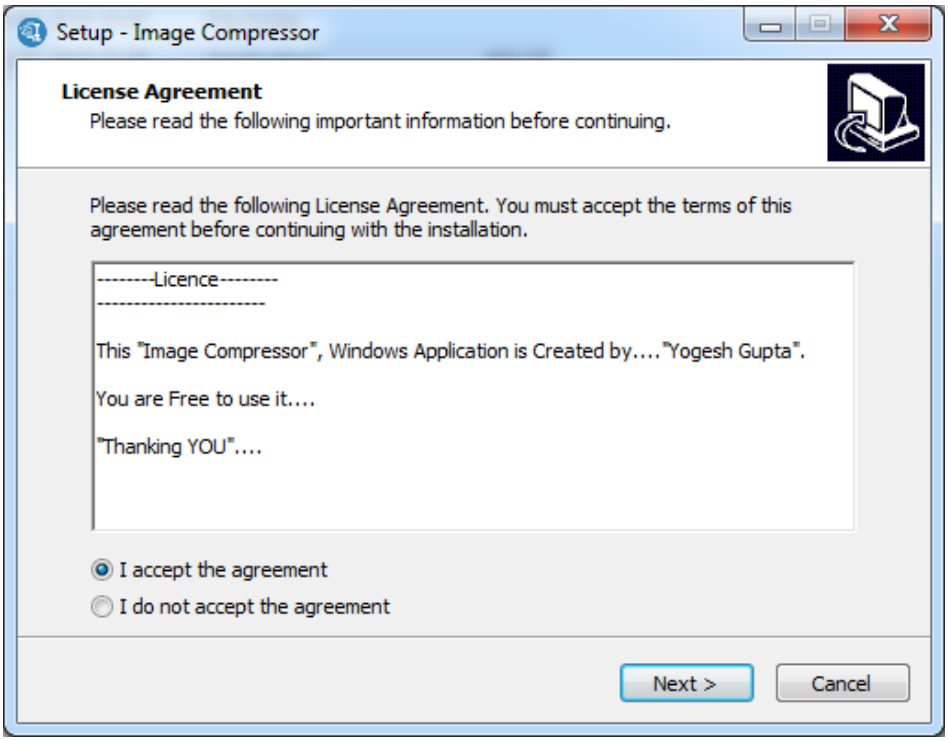
Step 2: Search for Download in the search area

Step 3: Click on Image Compressor where Setup File is written

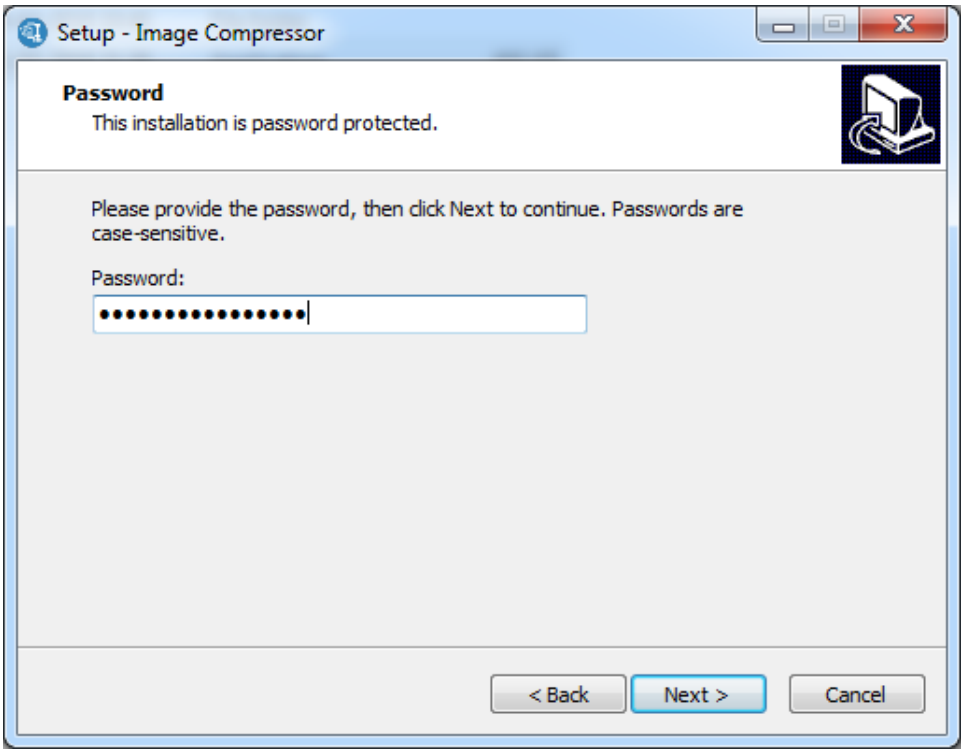
Step 4: Click on Download All to download the setup file and password file

The following steps guide you in installing the “Image Compressor” software.

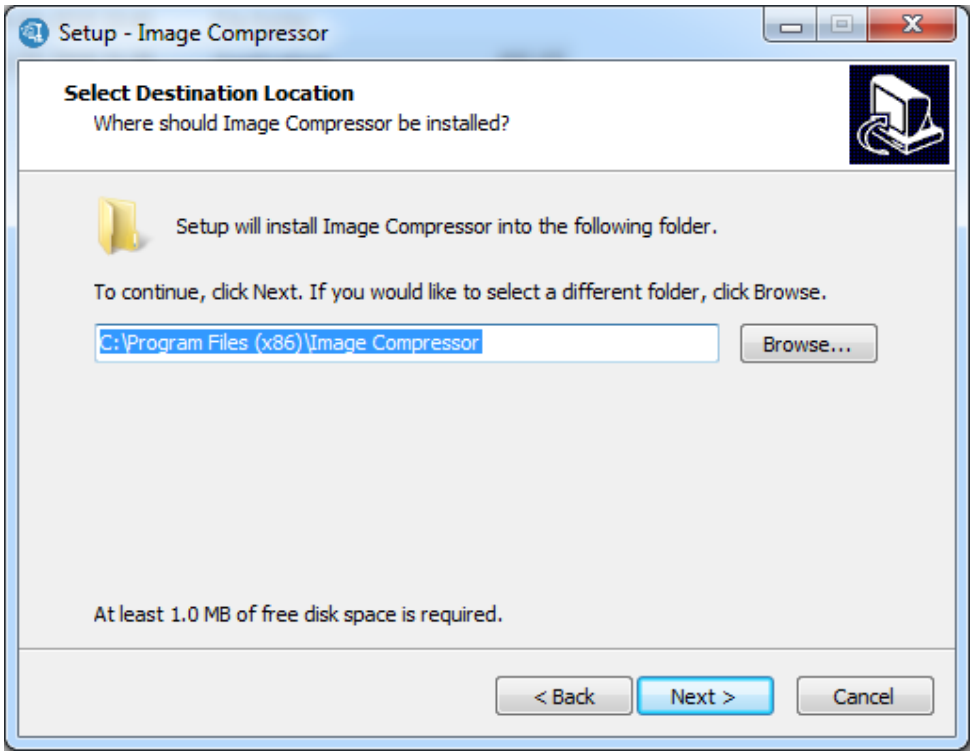
Step 1: Double click on the setup_icompressor, to run the installation click on Yes button and then select I accept the agreement and click on Next button



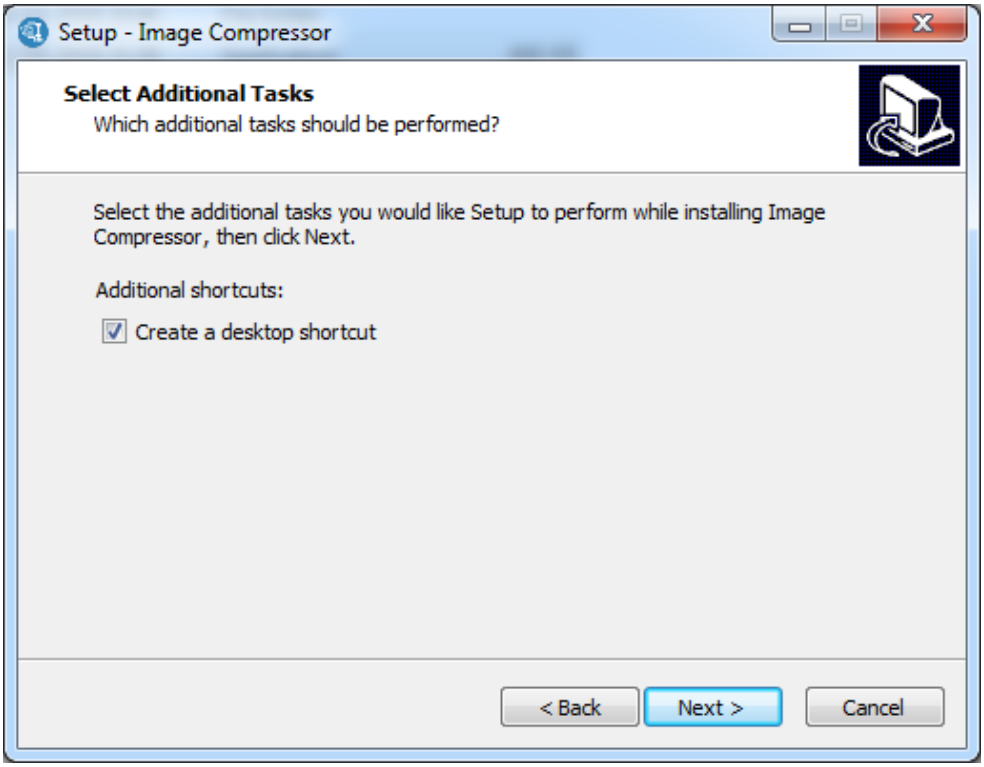
Step 2: Type the as given in the Password.txt file i.e. 4imagecompressor and click on Next button



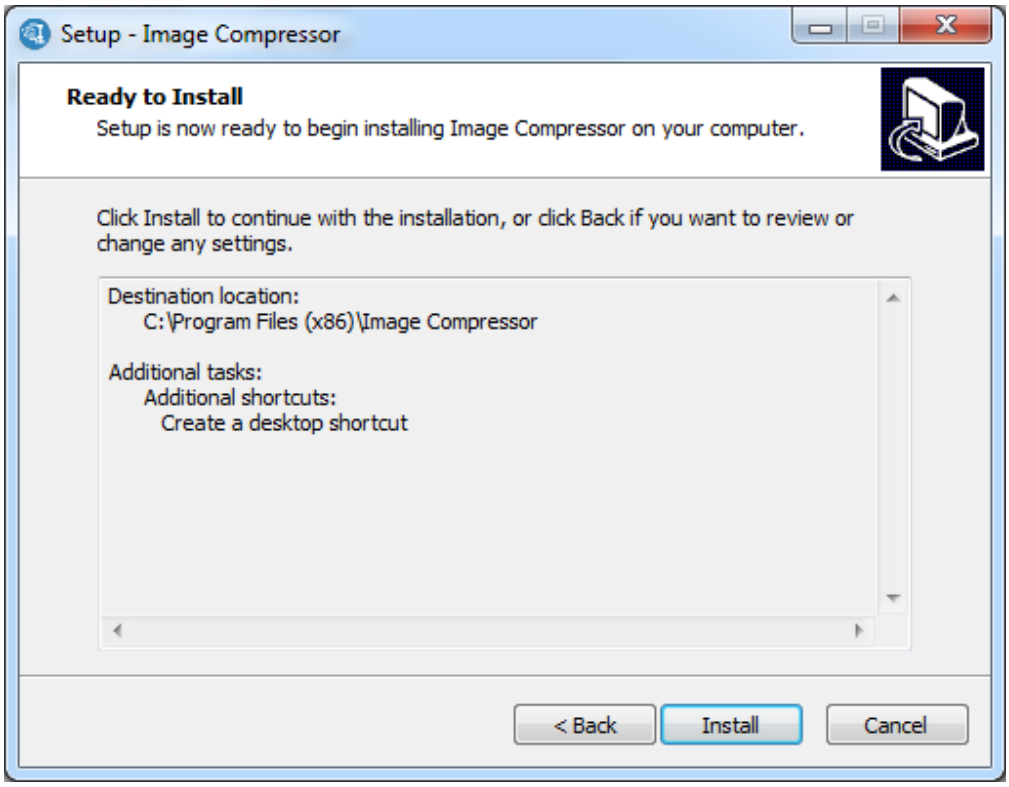
Step 3: Click on Next button on the Select Destination Location screen



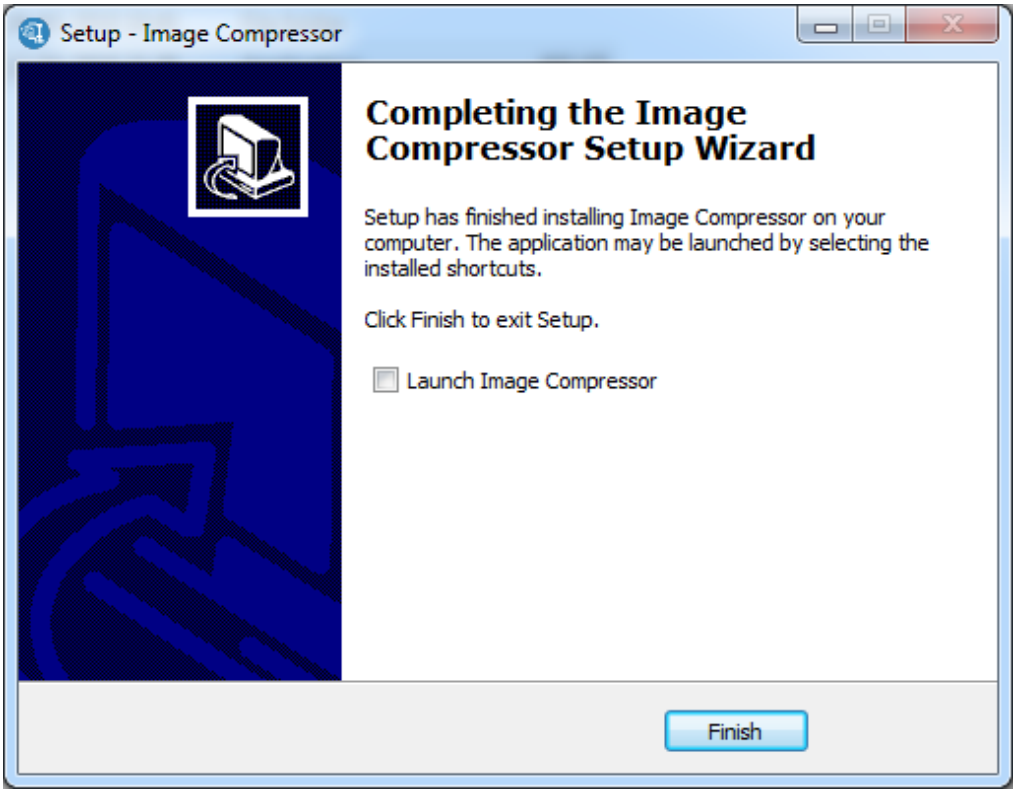
Step 4: On Select Additional Tasks screen check the checkbox Create a desktop shortcut and click on Next button



Step 5: Click on Install button on Ready to Install screen

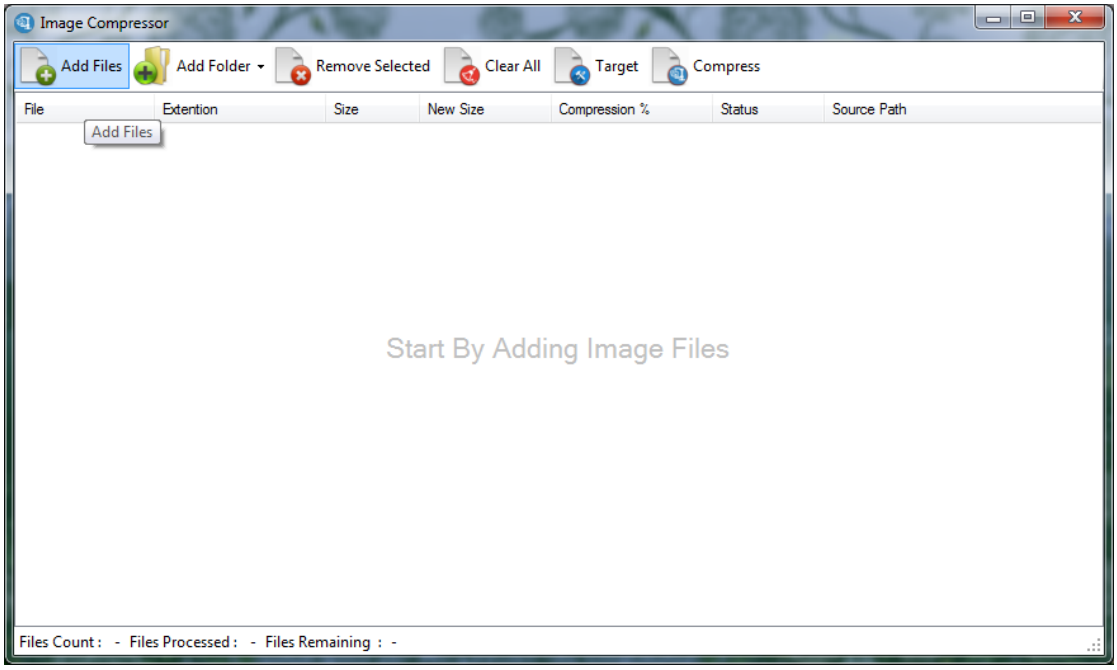


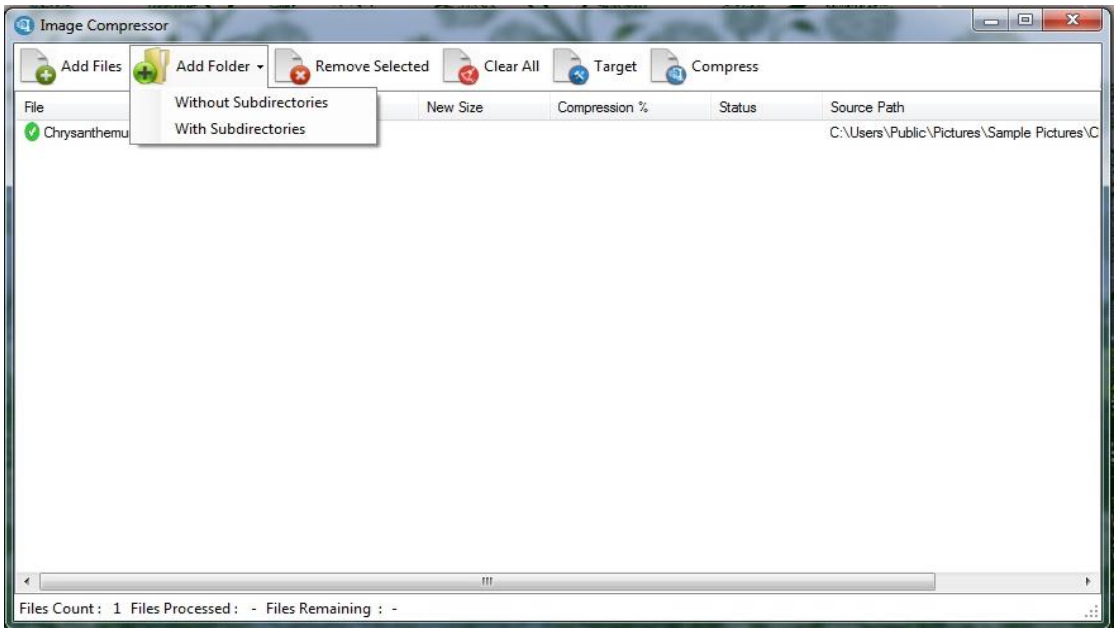
Step 6: Uncheck Launch Image Compressor on Completing the Image Compressor Setup Wizard and click on Finish button.

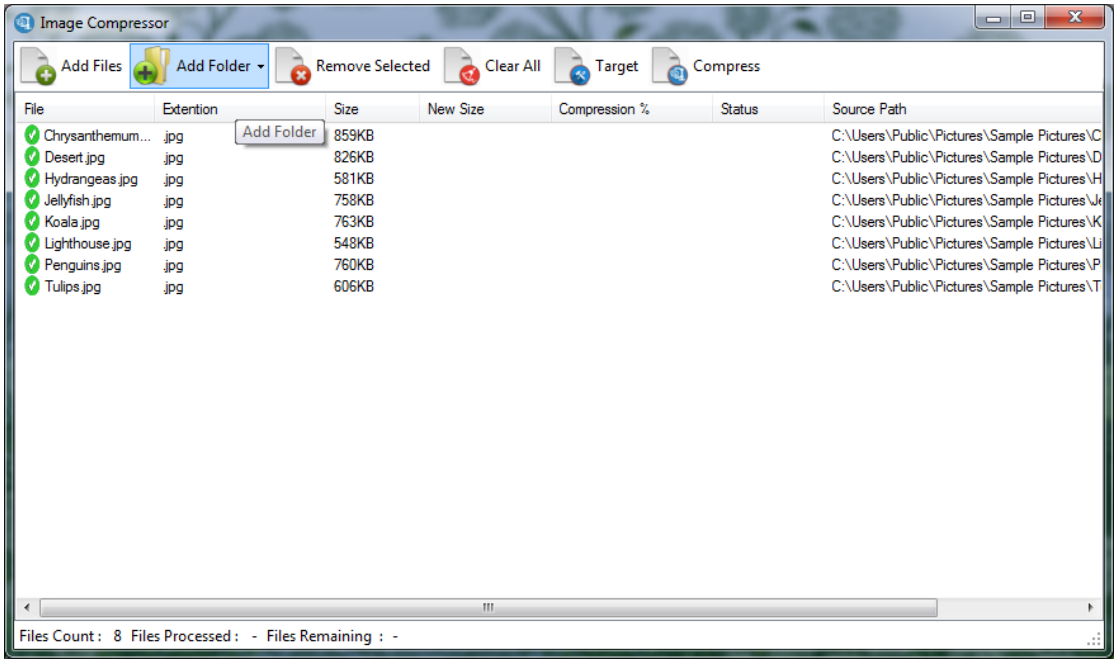


The following steps guide you in using the “Image Compressor” software.

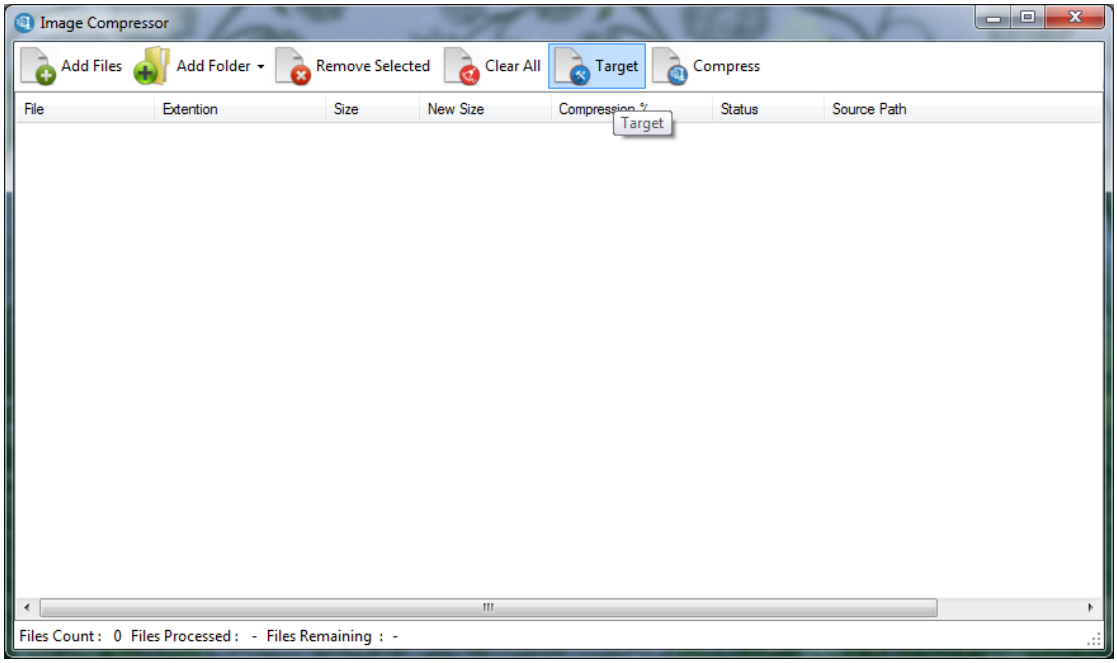
Step 1: Load images in the application by clicking on Add Files, Add Folder → With Subdirectory or Without Subdirectory button

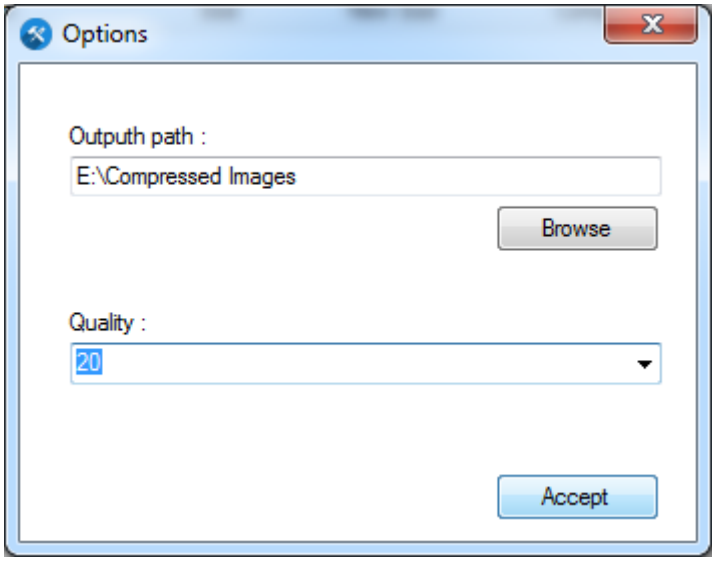






Step 2: To compress the images click on Target button and set Output path and Quality and then click on Compress button.




Options
Outputh path :
E:\Compressed Images
Browse
Quality :
20
Accept

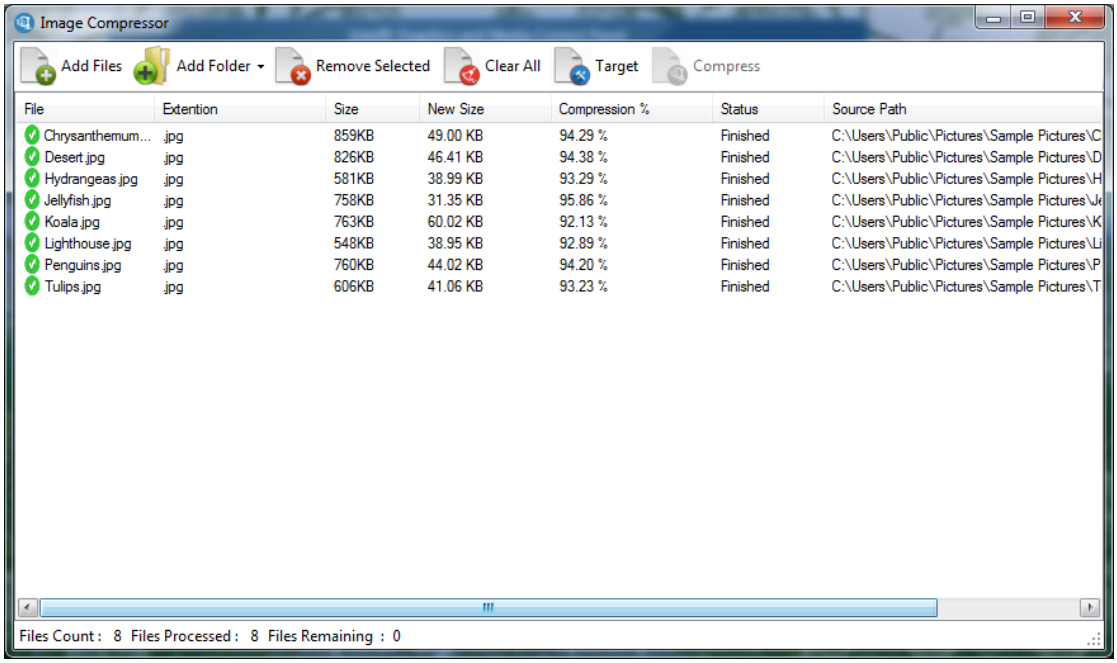
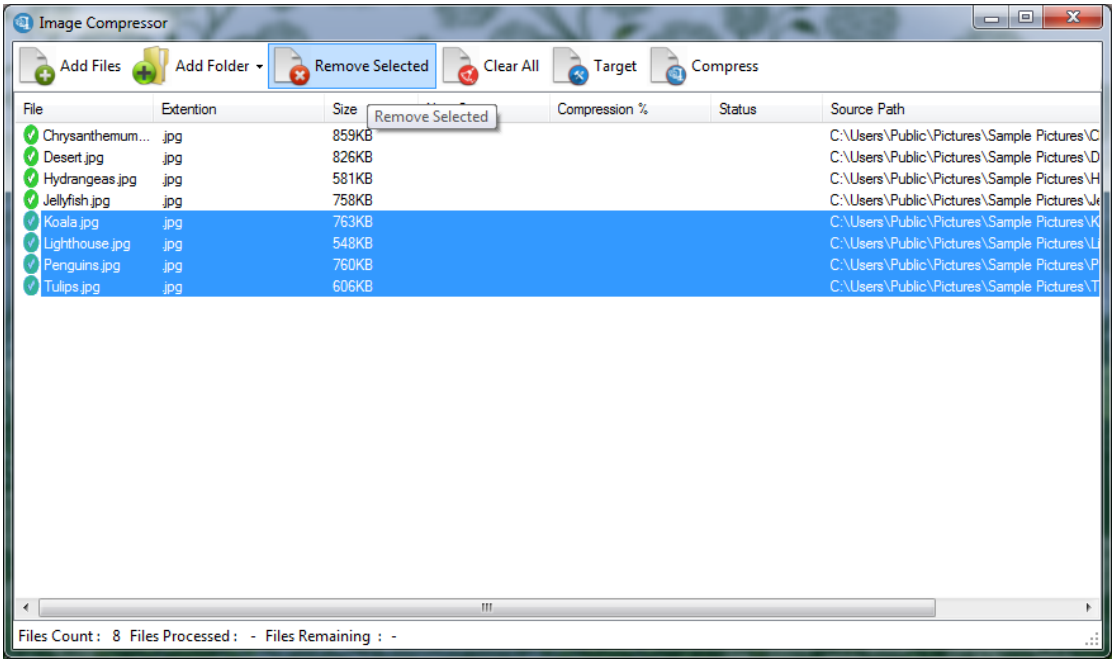
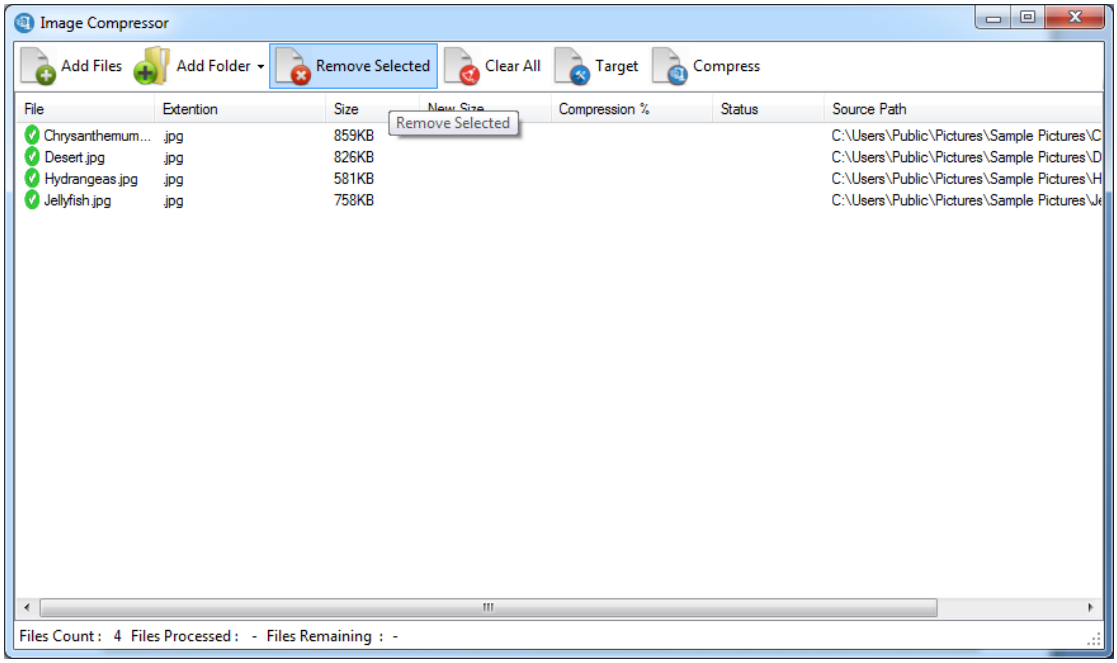


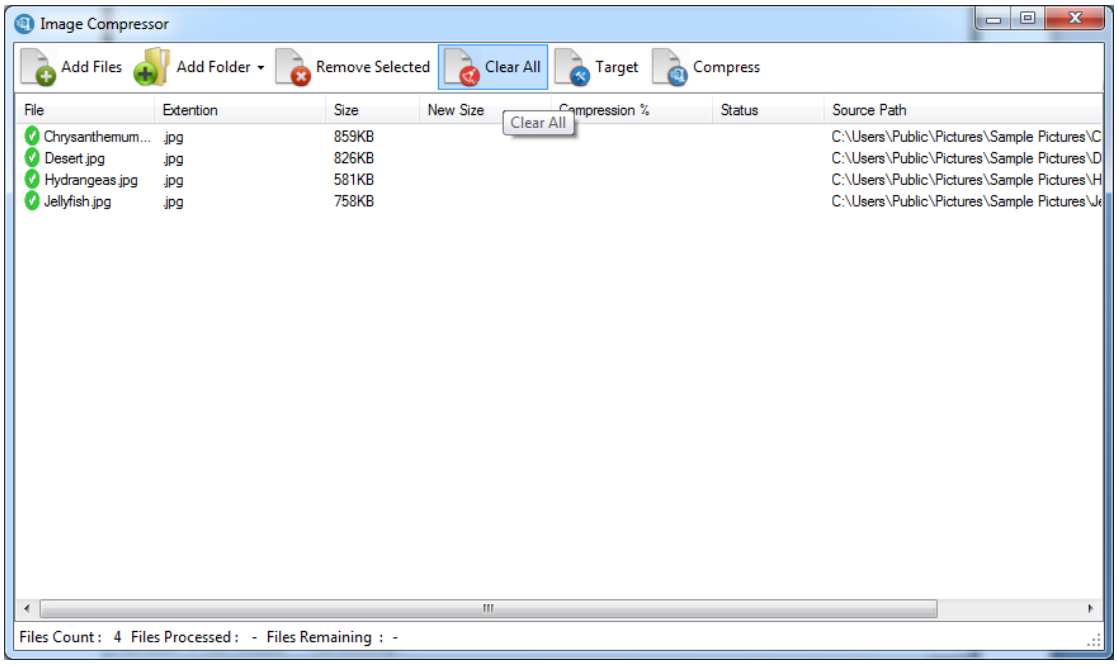
Image Compressor
Add Files
Add Folder
Remove Selected
Clear All
Target
Compress
File
Extention
Size
New Size
Compression %
Status
Source Path
Chrysanthemum... jpg 859KB 49.00 KB 94.29 % Finished C:\Users\Public\Pictures\Sample Pictures\C
Desert.jpg jpg 826KB 46.41 KB 94.38 % Finished C:\Users\Public\Pictures\Sample Pictures\D
Hydrangeas.jpg jpg 581KB 38.99 KB 93.29 % Finished C:\Users\Public\Pictures\Sample Pictures\H
Jellyfish.jpg jpg 758KB 31.35 KB 95.86 % Finished C:\Users\Public\Pictures\Sample Pictures\J
Koala.jpg jpg 763KB 60.02 KB 92.13 % Finished C:\Users\Public\Pictures\Sample Pictures\K
Lighthouse.jpg jpg 548KB 38.95 KB 92.89 % Finished C:\Users\Public\Pictures\Sample Pictures\Li
Penguins.jpg jpg 760KB 44.02 KB 94.20 % Finished C:\Users\Public\Pictures\Sample Pictures\P
Tulips.jpg jpg 606KB 41.06 KB 93.23 % Finished C:\Users\Public\Pictures\Sample Pictures\T
Files Count : 8 Files Processed : 8 Files Remaining : 0

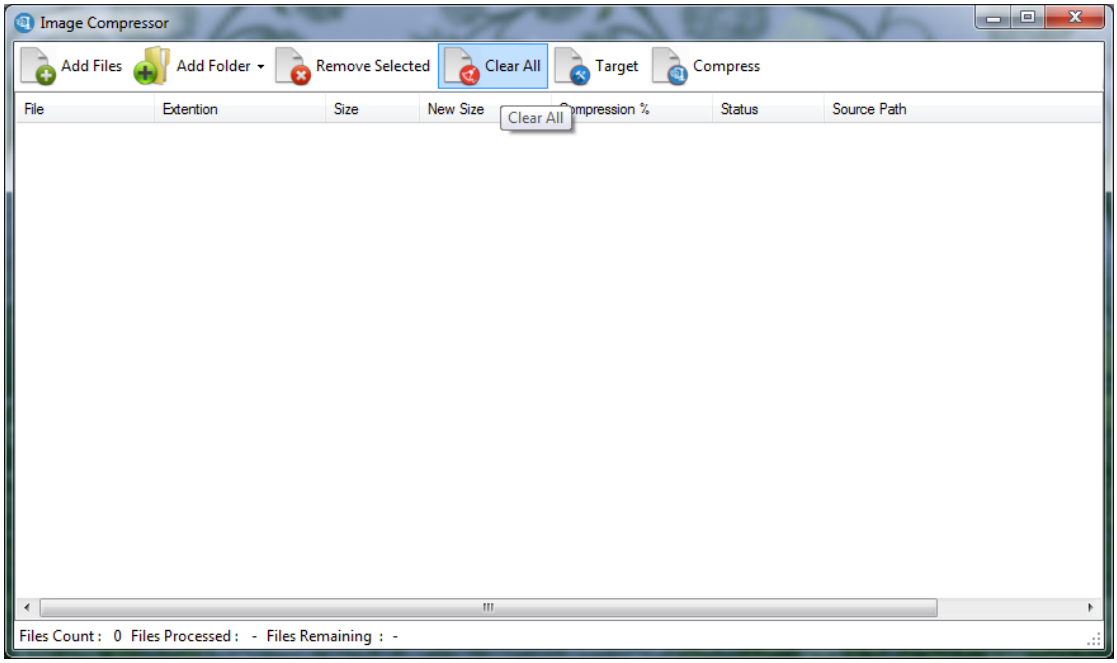
Step 3: To remove the unwanted images select the images and click on Remove Selected button





Step 4: To remove all the loaded images click on Clear All button





Abbreviations and Definitions

UI: The user interface (UI), in the industrial design field of human–computer interaction, is the space where interactions between humans and machines occur.

GUI: The graphical user interface (GUI), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

CPU: A central processing unit (CPU) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions.

Processor architecture: Nowadays, two most common types of processor are available32 bit (x86) and 64 bit (x64) which are based on 32 bit and 64 bit processor architecture. This affects the functionality and performance of the application.

Signing an Assembly: We can ensure an assembly's identity by signing it. This signing operation employs public-key infrastructure algorithms. Assembly manifests are hashed and signed with a private key. When any assembly is deployed, the hashed value is calculated and compared with the hashed value stored in the assembly. If the runtime hash value matches the hard-coded hash, the installation is allowed to continue.

.NET Reflector: .NET Reflector was the first CLI assembly browser. It can be used to inspect, navigate, search, analyze, and browse the contents of a CLI component such as an assembly and translates the binary information to a human-readable form.

Code Obfuscation: In software development, obfuscation is the deliberate act of creating source or machine code that is difficult for humans to understand.

References

1. Project and Product reference

- caesium-image-compressor, a software, can be found on the following link https://saerasoft.com/caesium/

2. Process reference

- UML Modeling for the Compression Of An Image File can be found on the following link https://pdfs.semanticscholar.org/33d6/2e3c30845e0ae70efb0e55de0a466e1feff7.pdf
- https://docs.microsoft.com/en-us/dotnet/framework/winforms/advanced/using-image-encoders-and-decoders-in-managed-gdi
- https://docs.microsoft.com/en-us/dotnet/framework/winforms/advanced/how-to-set-jpeg-compression-level
- https://docs.microsoft.com/en-us/dotnet/framework/winforms/advanced/using-transformations-in-managed-gdi

3. Synopsis reference

- "Society Pro", a society managing website project's synopsis made in second year of the B.Sc.I.T. academic year.
- Format for B.Tech Project Synopsis by Guru Nanak Dev Engineering College, Ludhiana on the following link https://www.gndec.ac.in/sites/default/files/Guidelines_for_Submission_of_Synopsis_of_Major_Project_and_Indutrial_Training_Report_of_Final_Year_IT_Students.pdf

4. Software requirement specification reference

- Cafeteria Ordering System, Release 1.0's SRS on the following link csis.pace.edu/~marchese/SE616_New/Samples/SE616_SRS.doc
- PeaZip Requirements for version 2.7.1's SRS on the following link www.peazip.org/documentation/SRS-PeaZip-2.7.1-EN.pdf

Acknowledgment

I had a great pleasure for representing this project report for **“Image Compressor”**, a windows application, and I grab this opportunity to convey my immense regards towards all the distinguished people who have their valuable contribution in the hour of need.

I profoundly thank our principal **“Dr. Vijay M. Sarode”** giving me support throughout the course and made me capable of being worthy of recognition and extended query facility to me for making and computing this project smoothly.

I would like to express my sincere thanks to **“Dr. Hiren Dand”**, Head of I.T. Department, for her constant encouragement, which made this project success.

I express my deep sense of gratitude to my project guide person for their timely encouragement, motivation and appraisal and for the solitude help and providing a sanctum atmosphere endeavouring to complete my project.

I also thank my family members for their continued support in completing this project work and last but not the least, I wish to thank all my friends and well-wishers who are directly or indirectly linked with the success of our project.