Topic: Image Compression

“To what extent is JPEG2000 a better image compression algorithm when compared to PNG and JPEG in terms of the tradeoff between the quality and the compression achieved?”

EXTENDED ESSAY IN COMPUTER SCIENCE

3997 words

Candidate code: gwj944

Table of Content

1. Introduction

As the digital imaging technology is evolving the clarity or in technical terms, the resolution of the images, had a steep increase in past few years, and it's still increasing. Better resolutions have resulted in larger file sizes. For example the recently commercialized resolution[1] i.e. 8K, has dimensions of 7680×4320, an uncompressed 8K image can be of nearly 95 MB. An image of 95MB may not consume much of resources at a smaller scale, but if looked at it on a larger scale, millions and millions of such 8K images, would consume a lot of resources while being transferred over a network(in terms of bandwidth utilization) or being stored at some server. To overcome this problem, many image compression algorithms have been developed by different organizations/computer scientists. These algorithms are capable of compressing images to smaller file size with/without affecting the quality of images, this completely depends on the image and the algorithm chosen.

In this extended essay I would be exploring how efficient JPEG2000 is, when compared to most widely used image compression algorithms i.e. JPEG and PNG using different type of images.

1.1. What are digital images?

In today's world, all the images are clicked on a electronic device. Once a image has been clicked, it has to be stored somewhere. As it has been clicked on an electronic device the most suitable option is to store it in digital format in a storage device (Pendrive, hard disk etc).

[1] "Dell's UP3218K 8K monitor is stunningly ahead of its time - The Verge." 12 Sep. 2017, https://www.theverge.com/2017/9/12/16293326/dell-8k-monitor-review-ultrasharp-up3218k. Accessed 10 Nov. 2017.

Digital images are made up of elements called pixels[2]. A pixel can be represented in different formats like binary, hex, rgb etc. A digital image is organised in a MxN array format, where M is the number of columns and N is the number of rows. Every coordinate (x, y) in this array is a pixel. The size of a uncompressed digital image completely depends on the dimensions of this array.

1.2. Why do we need digital images?

Digital images have made it possible to store huge number of images without using much of physical resources (In comparison to printing the images and “storing” them). Digital images have created so many new fields in the world of computer science, for example image recognition. Online application for anything conceivable is available today, this might require the user to upload certain documents online. This is enabled through the scanning of the documents and also it is legalised way to do so. Also, to display images on a website they need to be in digital format. These are just few example uses of digital images.

1.3. About compressing images to suit our requirements

Each compression algorithm has its own advantages and disadvantages, because of this, a user has to select a particular algorithm based on his/her requirements. A compression method is chosen on various factors. Few of them being storage availability, bandwidth (when the files have to be transferred over some network) and mainly, the purpose of the picture. An image clicked for some artistic purpose has to be of higher clarity, because of the very simple reason

[2] "All About Digital Photos - What Is a Digital Photo - Rideau-Info." http://www.rideau-info.com/photos/whatis.html. Accessed 12 Nov. 2017.

that the image’s detail matter more than the image’s size in this case, so the algorithm chosen over here, even though it doesn’t offer a high compression ratio, should not degrade the clarity/quality of the image. On the other hand if an image has been clicked just to keep it as a memory then it can be compressed to make the file’s size smaller, as the clarity of image does not matter much over here, algorithms that have a higher compression ratio even though they compromise with clarity is acceptable in this scenario.

2. Data Compression

Data compression generally means making a file’s size smaller. Any type of digital content (audio, video, web pages etc.) can be compressed using different compression algorithms, for example to compress video files MPEG-4[3] AVC is used. There are different type of algorithms which serve different file types, but the scope of this extended essay is focused towards image compression algorithms.

2.1. What is image compression?

Images have become an essential need nowadays. A huge chunk of bandwidth[4], while transferring over a network, and storage is used by images and so, it has become an essential need to compress them while storing or transmitting them. There are different kinds of image compression algorithms and all of them work in a more or less similar way, which will be explored in the next sub-topic. Image compression algorithms are of two types, lossy and

[3] "MPEG-4, Advanced Video Coding (Part 10) (H.264)." 21 Feb. 2017, https://www.loc.gov/preservation/digital/formats/fdd/fdd000081.shtml. Accessed 12 Nov. 2017.
[4] "Reducing Your Website's Bandwidth Usage - Coding Horror." 5 Mar. 2007, https://blog.codinghorror.com/reducing-your-websites-bandwidth-usage/. Accessed 13 Nov. 2017.

lossless. Lossy degrades the quality of the image but offers a high compression ratio while lossless does not compromise with the quality and therefore offers a low compression ratio, both of them have their own advantages and disadvantages. [5]

2.2. How does image compression work?

Image is compressed by removing/lessening different types of redundancies, in technical terms compression is achieved by reducing the total number of bits required to represent the image. This can be done by altering the images at pixel level. Generally, there are 3 basic types of redundancies :-

1. Inter-pixel redundancy[6]: Most of the time, there's a correlation between the neighbor pixels (adjustent pixels) and the observed pixel. The correlation can be predicted between them using value of the observed pixel.
2. Psycho-visual redundancy[6]: It is related to the sensitivity of our eyes in analyzing the image. Eliminating some less relative information is acceptable by human visual system.
3. Coding redundancy[6]: This redundancy is related to inefficient way of compiling data.

2.3. Types of image compression

Compression for any type of file can be classified into two categories lossy and lossless. The same goes for image compression. The two categories have been discussed below.

---

[5] "Everything You Need to Know About Image Compression - noupe." 3 Feb. 2010, https://www.noupe.com/design/everything-you-need-to-know-about-image-compression.html. Accessed 29 Nov. 2017.

[6] "Chapter 2 Digital Image Compression." http://digital.cs.usu.edu/~xqi/Proposal/Chapter2.pdf. Accessed 29 Nov. 2017.

2.3.1. Lossy compression algorithms

Lossy compression algorithms approximate the value of the data before compressing, which results in partial data loss, because of this they are also known as irreversible compression. The main motives behind using such algorithms is to reduce the redundancies and therefore, lessen the amount of resources used. The compression ratio offered by lossy algorithms is much higher compared to lossless algorithms.

2.3.1.1. What is lossy image compression algorithms?

Lossy image compression algorithms are useful for natural or real world images like photographs, these type of images contain complex edges and different colour gradients, not straight edges or areas of exactly one colour.

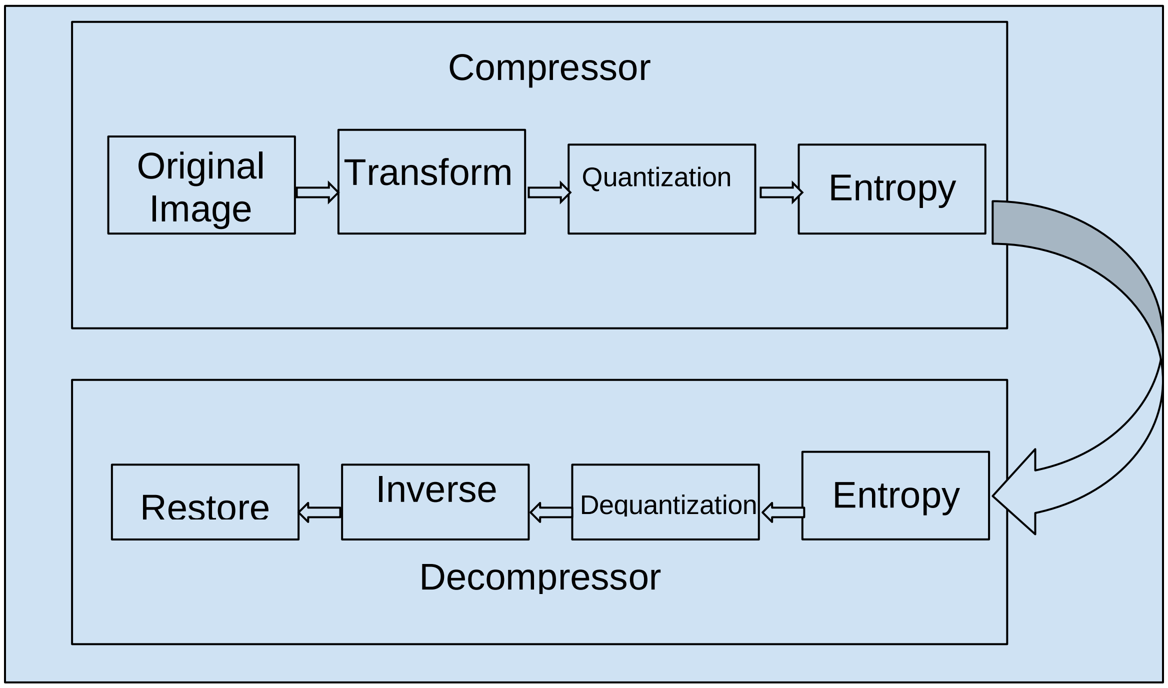


Figure 2.1. Lossy compression workflow[7]

[7] "Chapter 2 Digital Image Compression." http://digital.cs.usu.edu/~xqi/Proposal/Chapter2.pdf. Accessed 29 Nov. 2017.

Figure 2.1 is the diagrammatic representation of the basic workflow of a lossy image compression algorithm. In the first stage, i.e transform, the inter-pixel redundancy is reduced by packing the information efficiently. After this stage the image goes through a quantizer in which the psycho-visual redundancy is reduced/eliminated which in turn reduces the amount of data/bits required to represent the packed information. The quantized chunk of data is then efficiently encoded to get the best compression ratio which reduces the coding redundancy. The decompression works in exactly the inverse/opposite way of compression.

2.3.1.2. Methods used in lossy image compression algorithms.

Every lossy image compression algorithm follow the same flow as represented in figure 2.1. The thing that makes the difference is the different type of quantization techniques used.

Quantization is a way to map a set of pixel values with only one representative value. The two basic types of quantization are[8]:

1. Scalar quantization

   This quantization process is usually denoted as Y=Q(x). In this process the quantization function **Q** is used to map a scalar (one-dimensional) input value **x** to a scalar output value **Y**. This process maps each pixel. Scalar quantization can be as simple as rounding of a high-precision number to the nearest number, or to nearest multiple of some other unit precision.

[8] "4. Quantization and Data Compression - Purdue Engineering." https://engineering.purdue.edu/~ipollak/ece302/SPRING12/notes/25_Quantization_and_Compression.pdf . Accessed 1 Dec. 2017.

2. Vector quantization

   This quantization technique is derived from signal processing. Rather than quantizing every pixel value, vector quantization makes blocks of pixels and turns it into vectors after which it replaces the vector with a index in the codebook, whichever is the closest to the input vector using some closeness detecting algorithms. To decode this, the decoder simply looks into the codebook and replaces the index with the vector.

2.3.2. Lossless

Lossless algorithms don’t approximate any values while compression, because of which there is no loss of quality while compressing the image therefore, the original image can be recovered from the compressed image. Lossless algorithms are used when it is important to recover the exact file. As the uncompressed data has to be exact, the compression ratio is much lesser than lossy algorithms.

2.3.2.1. What is lossless image compression algorithms?

Lossless image compression algorithms are usually two step algorithms(figure 2.2). They are used in cases where even a minor detail of the image matters, for example medical imaging where even slightest loss in the image can cost a human life. These type of algorithms work well with images that contain lot of same color or straight lines. The decompresor works in exactly reverse way of the compressor.

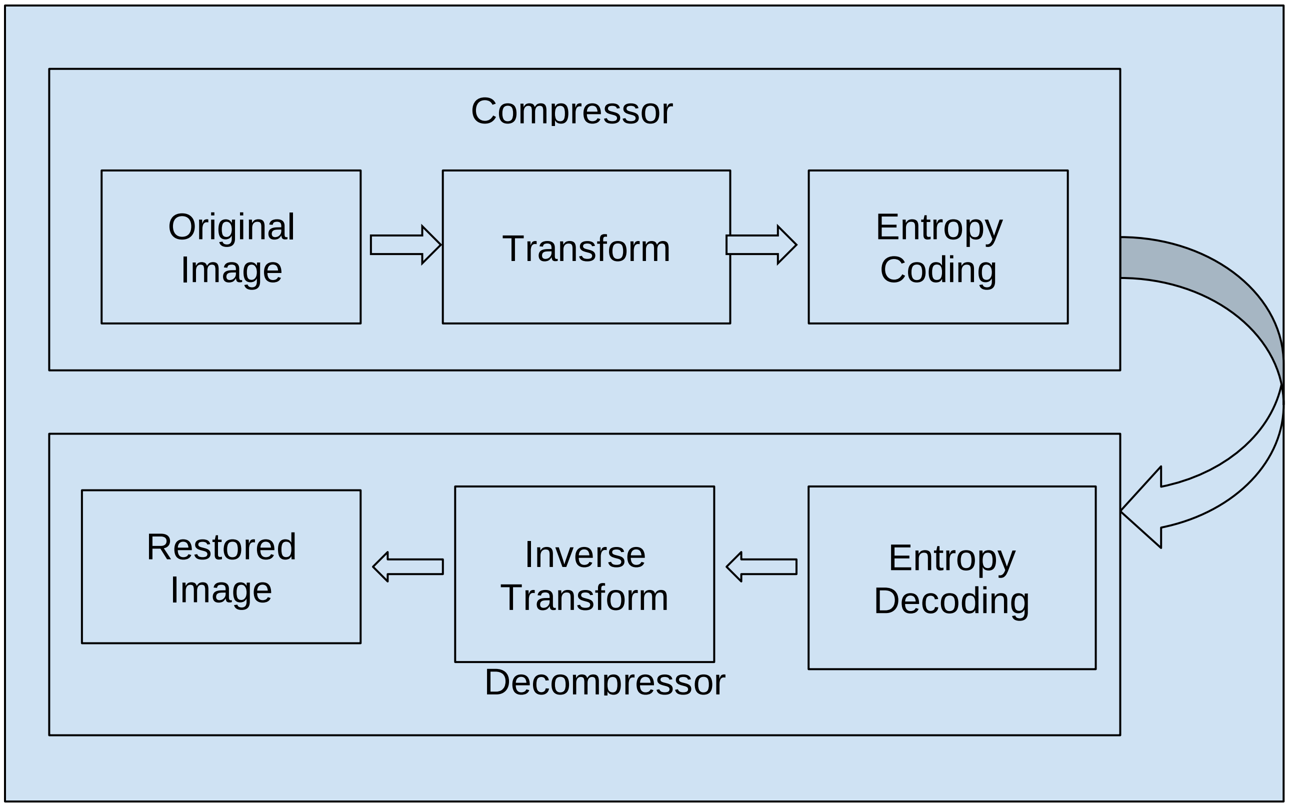


Figure 2.2. Lossless compression workflow[9]

The only difference between lossless and lossy compression algorithms is that lossless image compression algorithm do not go through any quantization(as shown in figure 2.2), because of which there's no data loss. The transform stage reduces the inter-pixel redundancy. In the second stage i.e entropy coding the bits (the pixels) are efficiently packed to reduce the coding redundancy.

[9] "Chapter 2 Digital Image Compression." http://digital.cs.usu.edu/~xqi/Proposal/Chapter2.pdf. Accessed 2 Dec. 2017.

2.3.1.2. Techniques used in lossless image compression

All the lossless compression algorithms follow the same flow as depicted in figure 2.2. The thing that makes the difference is different ways of encoding used in entropy coding stage. Three major lossless compression techniques are:

1. Run Length Encoding[10]:

   It replaces a series of identical pixels with the number of times they have occurred, for example:

   Original Sequence:

   2222111133333

   Is encoded to:

   (2,4)(1,4)(3,5)

2. Huffman Encoding[11]:

   Huffman coding replaces a set of common pixels with a short code word. The codes are stored in a codebook which can be constructed for a single or multiple images. The codebook has to be supplied to the decompressor to decode the codewords.

3. LZW Coding[12]:

[10] "Data Compression with Run-length Encoding - stoimen's web log." 9 Jan. 2012, http://www.stoimen.com/blog/2012/01/09/computer-algorithms-data-compression-with-run-length-encoding/. Accessed 2 Dec. 2017.
[11] "Greedy Algorithms | Set 3 (Huffman Coding) - GeeksforGeeks." http://www.geeksforgeeks.org/greedy-algorithms-set-3-huffman-coding/. Accessed 3 Dec. 2017.
[12] "LZW (Lempel–Ziv–Welch) Compression technique - GeeksforGeeks." http://www.geeksforgeeks.org/lzw-lempel-ziv-welch-compression-technique/. Accessed 4 Dec. 2017.

Lempel-Ziv- Welch(LZW) is a dictionary based encoding. The dictionary here can be either dynamic or static. It works in the same way to huffman, rather than replacing a set of common pixels LZW looks for recurring patterns throughout the image and replaces it with one keyword.

3. Image formats

There are many formats in which digital images are saved in. All formats have their own unique extensions. I have chosen 4 formats/algorithms, i.e BMP, PNG, JPEG and JPEG2000 for my extended essay.

3.1. BMP[13]

3.1.1. What is BMP image format?

BMP images are also known as bitmap images or device independent bitmaps (DIB). The extension for this image format is .bmp or .dib.

3.1.2. What are the characteristics of BMP images?

There is no compression with this image format. This allows the images to be of very high quality but it also results in extremely large file sizes.

[13] "BMP (Bitmap) Definition - The Tech Terms Computer Dictionary." https://techterms.com/definition/bmp. Accessed 4 Dec. 2017.

3.1.3. What’s the purpose of BMP images?

BMP images are used when the quality of image is much more important than the resources used while storing or transferring the image on a network. Image designers prefer to use BMP format because it allows them to alter every pixel of the image. These format is also used when very realistic images/graphics have to created.

3.2. PNG[14]

3.2.1. What is PNG image format?

PNG stands for Portable Network Graphics. It is one of the most widely used lossless image compression algorithm over the internet. This format was made to use as less bandwidth possible while transferring the file over internet. The extension for this file format is .png.

3.2.2. The characteristics of PNG

PNG images support transparent background, which means that while developing a image in PNG format a user can choose to have a transparent background. The transparent background is represented as black and white checkboxes in a image viewer. PNG only works with RGB(Red, green and blue) color schemes and therefore it does not support non-RGB color schemes, for example CMYK (cyan, magenta, yellow, and key (black)) color mode.

[14] "PNG (Portable Network Graphics) Home Site - libpng.org." 4 Nov. 2017, http://www.libpng.org/pub/png/. Accessed 5 Dec. 2017.

3.2.3. The purpose of PNG

PNG is used because of its very efficient lossless image compression and also because of its transparency feature.

3.4. JPEG

3.4.1. What is JPEG image format?

JPEG is an abbreviation of of Joint Photographic Experts Groups.It’s a lossy compression algorithm. This algorithm was designed to compress grayscale images or images with 24 bits dept. A human’s visual system is much more sensitive towards some factors compared to others, and this is what JPEG works on. Two such visual effects are[15]:

1. Human eyes are comparatively more sensitive towards brightness than to the chrominance.
2. Changes in homogeneous area attract more attention than areas where there is a lot of variations of colors.

The main reason why JPEG is such an efficient compression algorithm is because of its DCT quantization method[16]. DCT stands for discrete cosine transform. Before applying DCT on the image, the image gets converted to YCbCr color scheme from whichever color scheme the image is in. The Y component contains the brightness information and the grayscale version of the original image, other two components Cb and Cr contain the rest of color information. All three

[15] "How does JPEG actually work? – Developing for Developers." 12 Apr. 2006, https://blogs.msdn.microsoft.com/devdev/2006/04/12/how-does-jpeg-actually-work/. Accessed 5 Dec. 2017.
[16] "Jpeg Image Compression Using Discrete Cosine Transform - A ... - arXiv." https://arxiv.org/pdf/1405.6147. Accessed 5 Dec. 2017.

components altogether give the original image back. Once the image has been converted, each component is quantized differently. After this the components are broken into 8x8 blocks. DCT is applied onto each block. The resultant pixels are of always whole number, which means that the decimal values are ignored. The decompression works in completely the inverse way of the compression.The extensions of this format are .jpg/.jpeg/.jpe/.jif/.jfif/.jfi.

3.4.2. Characteristics of JPEG

The main defining characteristic of JPEG is that, it allows the users to decide the “quality” factor, in terms of percentage(1% to 100%). If the quality factor is high then the coefficients in quantization matrix will be more which will result in higher quality and bigger file size. Where as, if the quality factor is low then there will be less coefficients in the quantization matrix which will result in lower quality but small file size. [17]

3.4.3. Purpose of JPEG

JPEG is one of the most widely used lossy image compression algorithm for storing and transmitting images on internet. This compression method can work very efficiently on real life images, i.e photographs or anything that looks realistic(a realistic painting etc).

[17] "Lossy Data Compression: JPEG - CS Stanford." https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/coeff.htm. Accessed 5 Dec. 2017.

3.5. JPEG2000

3.5.1. What is JPEG2000 image format?

JPEG2000 was created by the same organization that created JPEG, Joint Photographic Expert Group, with an intention to supersede DCT(refer to 3.4.1) using a new transformation method i.e Discrete Wavelet Transform also known as DWT. This compression algorithm works on the same concepts as mentioned above (refer to 3.4.1). The major difference is the DWT and few tweaks in entropy coding scheme. The extension for this file format are .jp2, .jpf, .jpx, .jpm and .mj2

3.5.2. Characteristics of JPEG2000

JPEG2000 has a better compression ratios than JPEG. This is majorly due to the new method of transformation i.e DWT which stands for Discrete wavelet transform. Unlike jpeg, which offers just lossy compression, JPEG2000 offers both lossy and lossless compression. It's up to user's distinction to choose the compression type. The another major difference is that, JPEG2000 supports 16-bit & 32-bit unlike JPEG which just supports 8-bit color images.

3.5.3. Purpose of JPEG2000

As mentioned above, the main motive behind developing JPEG2000 was to give a better version of JPEG. According to the developers, JPEG offers much better compression ratio but compression and decompression takes much longer. As mentioned in a journal named "The JPEG2000 Image coding Standard" by Dr. Dobb's Journal; April 2001, the time taken to encode is roughly 3 times more and decoding time is nearly 8 times more than JPEG.

4. Methodology

4.1. Explanation

4.1.1. What type of images are going to be used?

I have decided to use 3 different types of images, i.e. Grayscale image (clicked from a high definition camera in grayscale mode and saved in BMP format), high definition image (clicked from a high definition camera and saved in BMP format) and scanned document (scanned from a scanner and saved in BMP format). As each of these algorithms have a different way of compressing images, I believe that an algorithm may outperform the rest of the algorithms in a particular type of image. This is main reason behind using different type of pictures.

4.1.2. Research Process

I have used Matlab (imwrite function[18]) to convert the image into the desired image format. All the algorithms that I am exploring (JPEG, JPEG2000 & PNG) in this essay are supported by this function. On top of that, it even has an option to pass the quality factor in JPEG and the compression ratio in JPEG2000. The problem is that JPEG takes quality as its parameter whereas JPEG2000 takes compression ratio as its parameter. So, as to make the comparison possible what I have planned to do is, convert the uncompressed images into JPEG by passing 10,20,30,40,50,60,70,80,90 and 100 as the quality parameters one by one, this will give me 10

[18] "Write image to graphics file - MATLAB ...." https://www.mathworks.com/help/matlab/ref/imwrite.html. Accessed 10 Dec. 2017.

different images after which I’ll calculate the compression ratio for each image by using the following formula:

Compression Ratio = Uncompressed Image Size / Compressed Image Size[19]

Once I get the compression ratio, I’ll pass that value into the compression ratio parameter of JPEG2000 because of which, I’ll be able to make a comparison in terms of the difference in the quality (PSNR value) of the image as the size of the file will more or less be the same.

To find the reduction in the quality of the image due to compression, in comparison with uncompressed image, I’ll be calculating the PSNR i.e Peak Signal-to-Noise Ratio of the compressed images. Even this function is provided by Matlab[20]. The PSNR is calculated using the following mathematical formula[21]:

$$PSNR = 20\log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\|f(i,j) - g(i,j)\|^2$$

Legend:

f represents the data (in matrix format) of the uncompressed image.

[19] "How to calculate image compression ratio? - MATLAB Answers ...." 10 Aug. 2015, https://uk.mathworks.com/matlabcentral/answers/237620-how-to-calculate-image-compression-ratio. Accessed 15 Dec. 2017.
[20] "Peak Signal-to-Noise Ratio (PSNR) - MATLAB psnr - MathWorks." https://www.mathworks.com/help/images/ref/psnr.html. Accessed 15 Dec. 2017.
[21] "Peak Signal-to-Noise Ratio as an Image Quality Metric - National ...." 11 Sep. 2013, http://www.ni.com/white-paper/13306/en/. Accessed 15 Dec. 2017.

g represents the data (in matrix format) of the compressed image.

m represents the numbers of rows of pixels of the images and i represents the index of that row

n represents the number of columns of pixels of the image and j represents the index of that column

MAX_f if the maximum signal value that exists in the uncompressed image.

PSNR basically tells about the numerical difference between the value of pixels of uncompressed image and the compressed image. The higher the PSNR value the more "identical" the compressed image is to the uncompressed image in terms of pixels, which can also be said as, the higher the PSNR value the better the quality of image. There is no point calculating PSNR for lossless compression algorithms because of the very simple reason that, there is no loss of data/quality in lossless algorithms.

Note: As all the images are more less the same, there is no point of displaying each one of them in the main essay. Therefore, all the images will be included in the appendix.

So, the flow of exploration for each type of image will be like this:

1. Displaying the original image with the image size with image dimensions.
2. Compressing the image using JPEG with 10 different values in "quality parameter" (i.e 10,20,30,40,50,60,70,80,90 and 100). After that for each image I'll be calculating the PSNR value and the compression ratio and then I will display it in the following format:

Quality parameter passed in JPEG	Compressed image size (Megabytes)	PSNR Value	Compression Ratio (original image's size/compressed image size in

			megabytes) [22]

3. Then comes the turn of JPEG2000. I'll pass each compression ratio into JPEG2000's "compression ratio" parameter, after this I will have 10 different JPEG images and 10 different JPEG2000 images. Now as the compression ratio is constant between two images(one of JPEG and one of JPEG2000), the size of those two images will be nearly the same. Because of this, I can ignore the image size factor and compare images based on the PSNR values.
4. Images will be converted in PNG and JPEG2000 lossless after which, I'll calculate the compression ratio.

Algorithm's Name	Compressed image size (Megabytes)	Compression ratio (original image's size/compressed image size in megabytes) [1]
JPEG2000 lossless		
PNG		

5. I'll plot the graph for lossless and lossy algorithms separately.

I am assuming the following things about the matlab functions (imwrite):

[22] Values are upto 4 digits after decimal points.

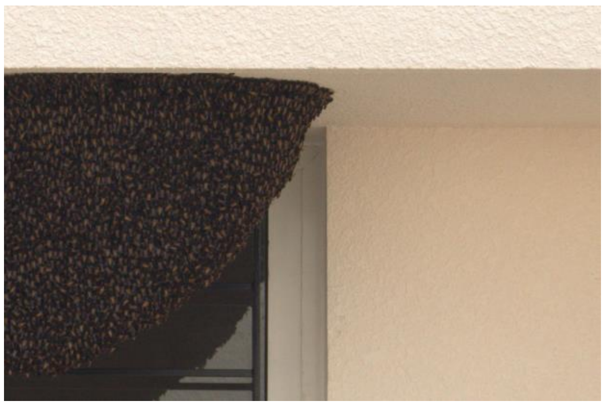
1. The imwrite function is up to date (according to the standards of those particular algorithms) and is free of any error.
2. The PSNR function of Matlab follows the exact mathematical formula as mentioned above.

This process will be repeated for each type of image. Therefore, I'll be able to analyze which algorithm works better with which type of image and make a conclusion based on my findings at the end of this extended essay.

4.2. Observation

4.2.1. Comparison using high resolution photographs

I clicked a picture of a beehive from a high definition camera and saved it in BMP format. I cropped a specific part of the hive, which was mainly focused towards the edges of the hive. I'll use this cropped image for the compression research. The main motive behind using this cropped picture is to see how efficiently these algorithms can deal with soft edges.

Original Image	Cropped Image
	

File size: 43.64 MB (2 d.p) Image dimensions: 4948 x 3280	File size: 5.16 MB (2 d.p) Image dimensions: 1610 x 1067

I compressed the original high resolution uncompressed image into 10 different variants of JPEG, after which, I found/calculated the following information:

Quality parameter passed in JPEG	Compressed image size (Megabytes)	PSNR Value	Compression Ratio (5.155798/compressed image size in megabytes) 4 d.p*
10	0.038544	30.1279	133.7640
20	0.05531	32.4143	93.2164
30	0.072445	33.5108	71.1684
40	0.089628	34.1343	57.5244
50	0.108049	34.5303	47.7172
60	0.129068	34.9479	39.9464
70	0.162988	35.4304	34.6330
80	0.218491	36.0652	23.5973
90	0.352218	37.0430	14.6381

100	1.357061	42.0475	3.7992

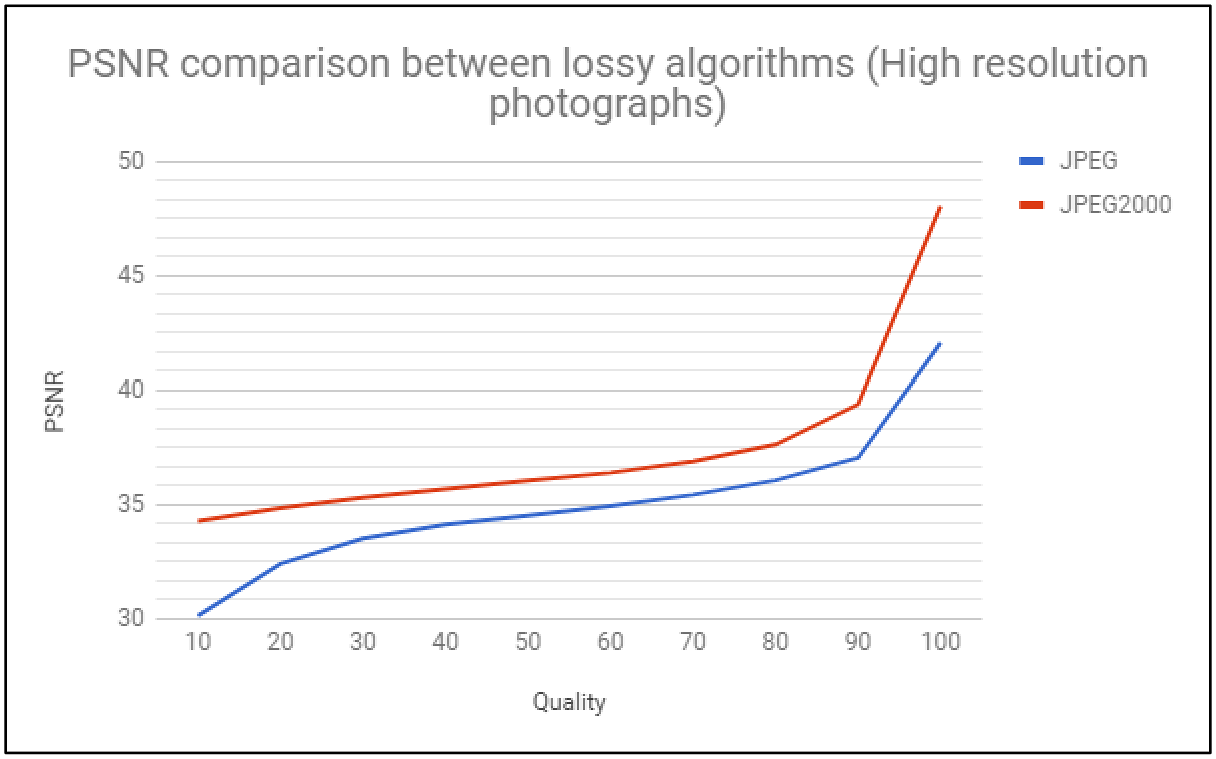
I passed the values of compression ratio in parameter of JPEG2000 to achieve the same scale of compression, the PSNR values after compressing the images are as follows:

Compression ratio calculated from the findings in JPEG	Quality parameter that was passed in JPEG to achieve that ratio	PSNR of image compressed using the JPEG2000 by passing the compression ratio
133.7640	10	34.284
93.2164	20	34.843
71.1684	30	35.3014
57.5244	40	35.6923
47.7172	50	36.0622
39.9464	60	36.394
34.6330	70	36.882
23.5973	80	37.628
14.6381	90	39.3784
3.7992	100	48.022

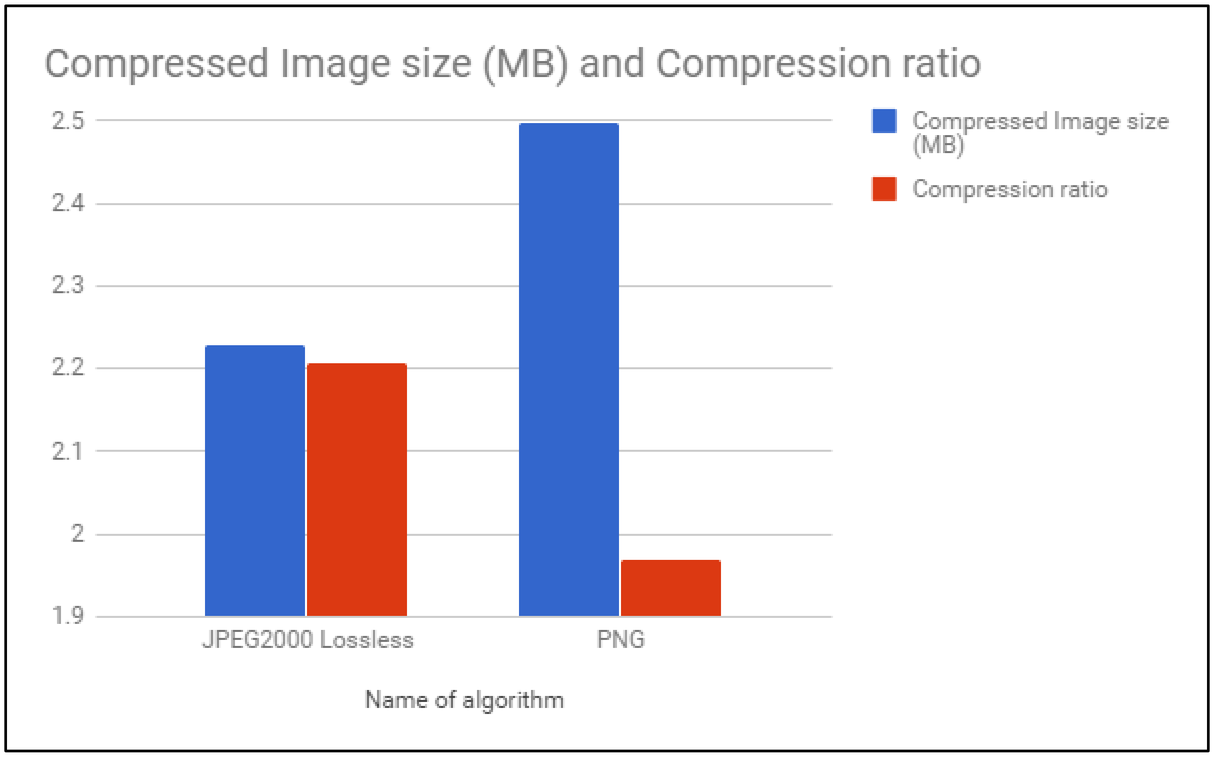
Compressing the original image into the lossless algorithms, JPEG2000 lossless and PNG, gave me the following results:

Algorithm's Name	Compressed image size (Megabytes)	Compression ratio (5.155798/compressed image size in megabytes) [1]
JPEG2000 lossless	2.2292	2.3128
PNG	2.4968	2.05

Graphical representation of the data found/calculated for lossy algorithms:



Graphical representation of data found/calculated for lossless algorithms:



4.2.2. Comparison using Grayscale images

To get a grayscale image, I set the camera's color scheme option to grayscale. After which, I clicked an image of a scenery.

Original Image

File size:46.2 MB

Image dimensions:4940 x 3272

I compressed the original grayscale raw image into 10 different variants of JPEG,after which, I found/calculated the following information:

Quality parameter passed in JPEG	Compressed image size (Megabytes)	PSNR Value	Compression Ratio (46.245295269/compressed image size in megabytes) 4 d.p*

10	0.27133	36.2957	170.4393
20	0.31853	39.7726	145.1801
30	0.37025	41.2944	124.9006
40	0.41861	42.1876	110.4713
50	0.47043	42.9497	98.3032
60	0.53047	43.4477	87.1778
70	0.64085	44.1062	72.1622
80	0.86527	44.933	53.4456
90	1.5612	46.3054	29.6221
100	6.3361	58.4847	7.2986

I passed the values of compression ratio in parameter of JPEG2000 to achieve the same level of compression, the PSNR values after compressing the images are as follows:

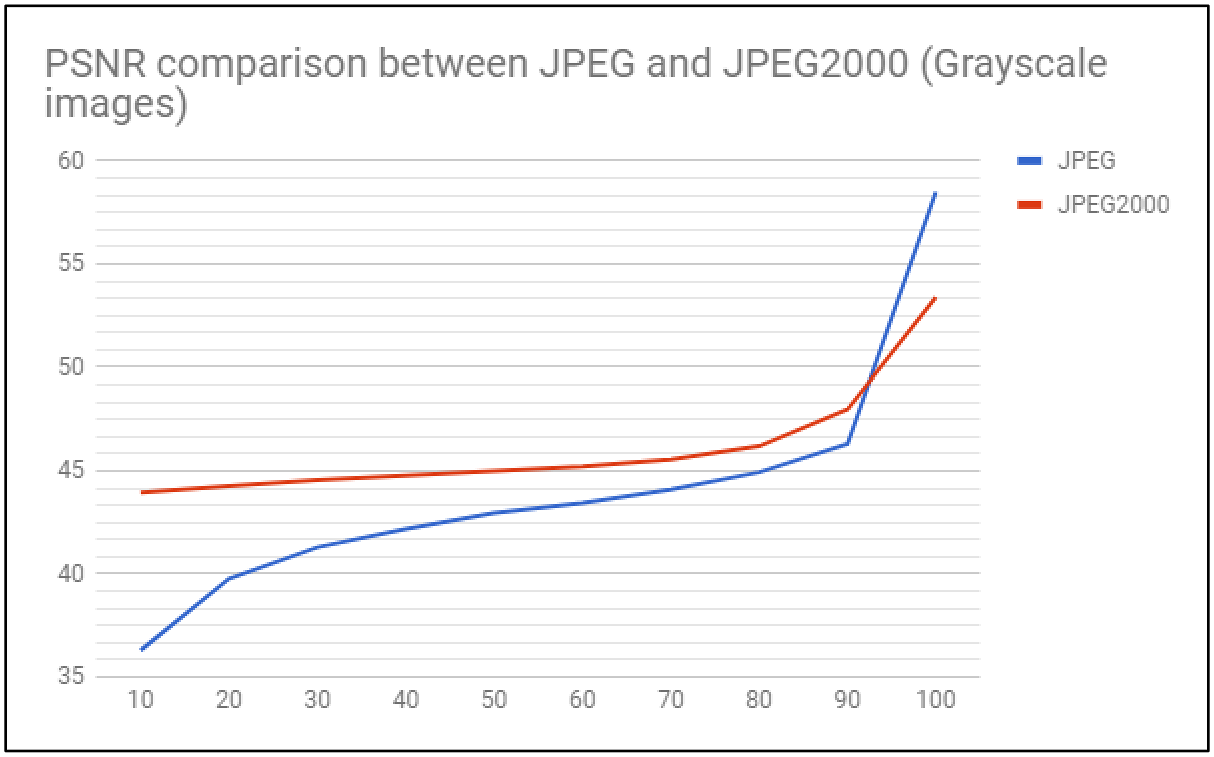
Compression ratio calculated from the findings in JPEG	Quality parameter that was passed in JPEG to achieve that ratio	PSNR of image compressed using the JPEG2000 by passing the compression ratio
170.4393	10	43.9429
145.1801	20	44.2764
124.9006	30	44.5598

110.4713	40	44.7839
98.3032	50	45.0028
87.1778	60	45.2173
72.1622	70	45.5612
53.4456	80	46.1976
29.6221	90	47.977
7.2986	100	53.3884

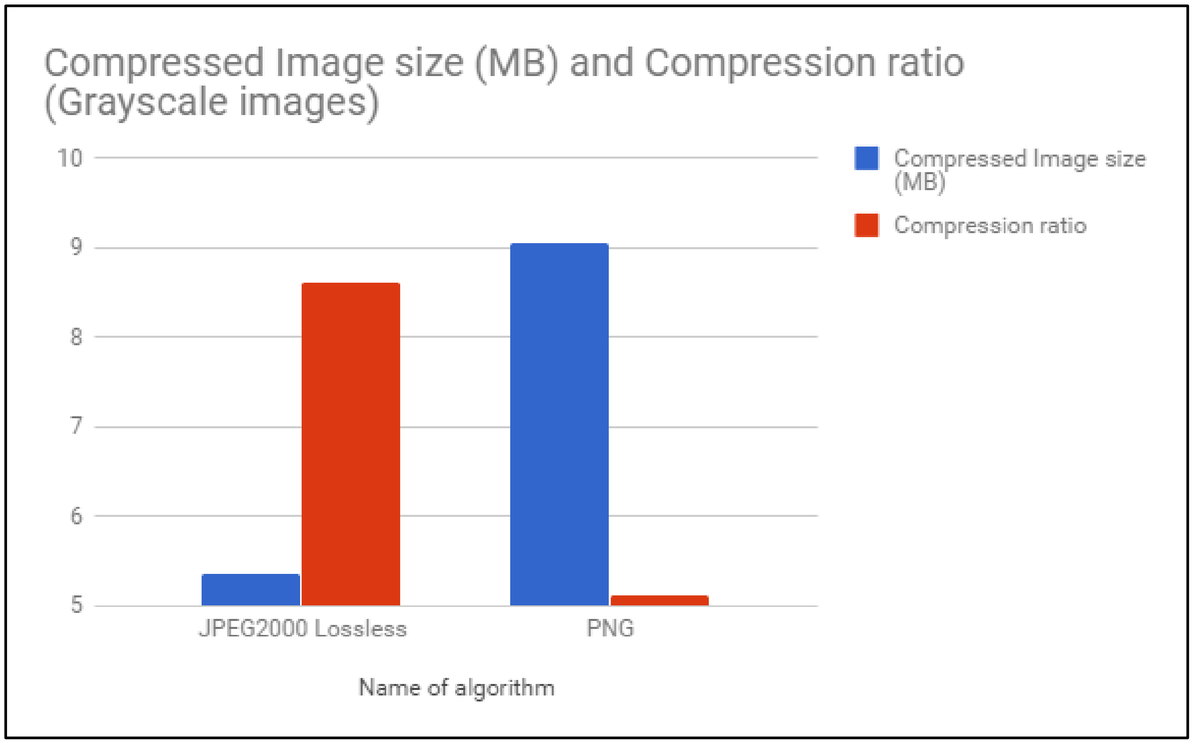
Compressing the original image with the lossless algorithms, gave me the following results:

Algorithm's Name	Compressed image size (Megabytes)	Compression ratio (original image's size/compressed image size in megabytes) [1]
JPEG2000 lossless	5.3657	8.6186
PNG	9.0518	5.1089

Graphical representation of the data found/calculated for lossy algorithms (i.e JPEG and JPEG2000):



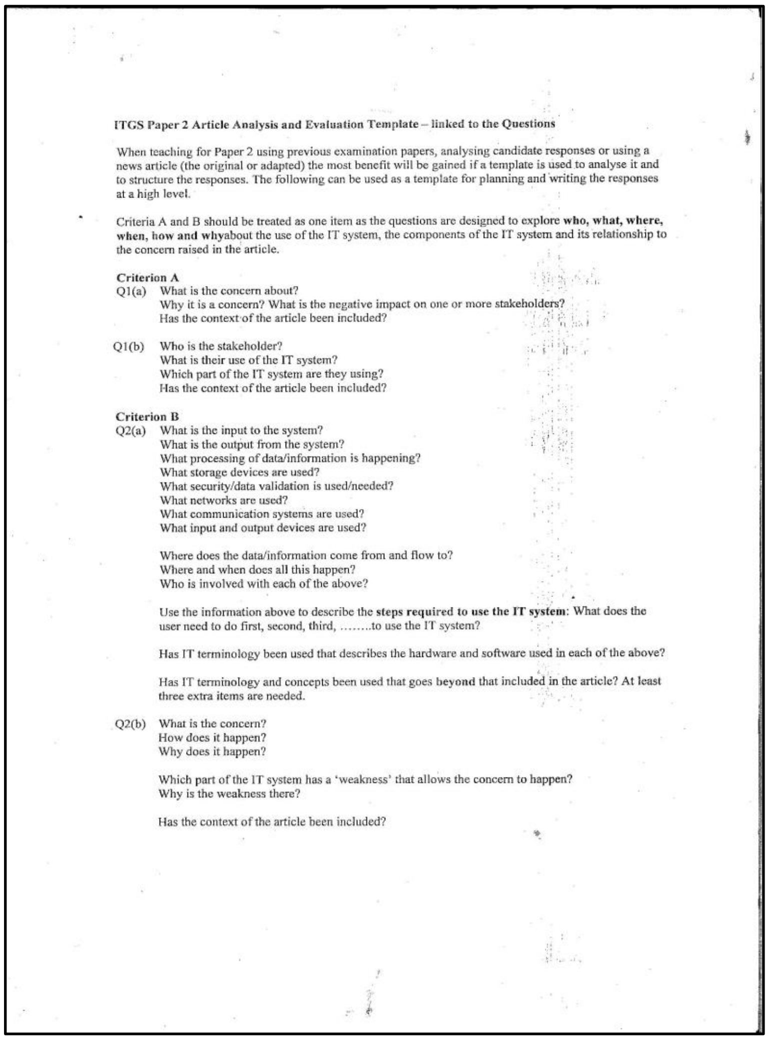
Graphical representation of data found/calculated for lossless:



4.2.3. Comparison using scanned documents

The school I study in has a commercial printer. This printer had an option to set the desired format of the output file of the scanned document. I set this option to BMP and scanned one of the documents I collected from my computer science teacher.

Original image



ITGS Paper 2 Article Analysis and Evaluation Template – linked to the Questions

When teaching for Paper 2 using previous examination papers, analysing candidate responses or using a news article (the original or adapted) the most benefit will be gained if a template is used to analyse it and to structure the responses. The following can be used as a template for planning and writing the responses at a high level.

Criteria A and B should be treated as one item as the questions are designed to explore **who, what, where, when, how and why**about the use of the IT system, the components of the IT system and its relationship to the concern raised in the article.

Criterion A

Q1(a) What is the concern about?
Why it is a concern? What is the negative impact on one or more stakeholders?
Has the context of the article been included?

Q1(b) Who is the stakeholder?
What is their use of the IT system?
Which part of the IT system are they using?
Has the context of the article been included?

Criterion B

Q2(a) What is the input to the system?
What is the output from the system?
What processing of data/information is happening?
What storage devices are used?
What security/data validation is used/needed?
What networks are used?
What communication systems are used?
What input and output devices are used?

Where does the data/information come from and flow to?
Where and when does all this happen?
Who is involved with each of the above?

Use the information above to describe the **steps required to use the IT system**: What does the user need to do first, second, third, ……..to use the IT system?

Has IT terminology been used that describes the hardware and software used in each of the above?

Has IT terminology and concepts been used that goes **beyond** that included in the article? At least three extra items are needed.

Q2(b) What is the concern?
How does it happen?
Why does it happen?

Which part of the IT system has a 'weakness' that allows the concern to happen?
Why is the weakness there?

Has the context of the article been included?

File size: 11.08 MB

Image dimensions: 1656 x 2339

I compressed the original high resolution uncompressed image into 10 different variants of JPEG,after which, I found/calculated the following information:

Quality parameter passed in JPEG	Compressed image size (Megabytes)	PSNR Value	Compression Ratio (11.08185396/compressed image size in megabytes) 4 d.p*
10	0.14477	25.3903	76.548
20	0.187	27.6019	59.2612
30	0.2255	30.2117	49.1434
40	0.25389	32.2548	43.6481
50	0.27939	34.1579	39.6647
60	0.30482	36.014	36.3553
70	0.33847	38.4054	32.7417
80	0.38587	41.8871	28.7192
90	0.48398	47.9836	22.8977
100	0.90287	71.4491	12.2742

I passed the values of compression ratio in parameter of JPEG2000 to achieve the same scale of compression, the PSNR values after compressing the images are as follows:

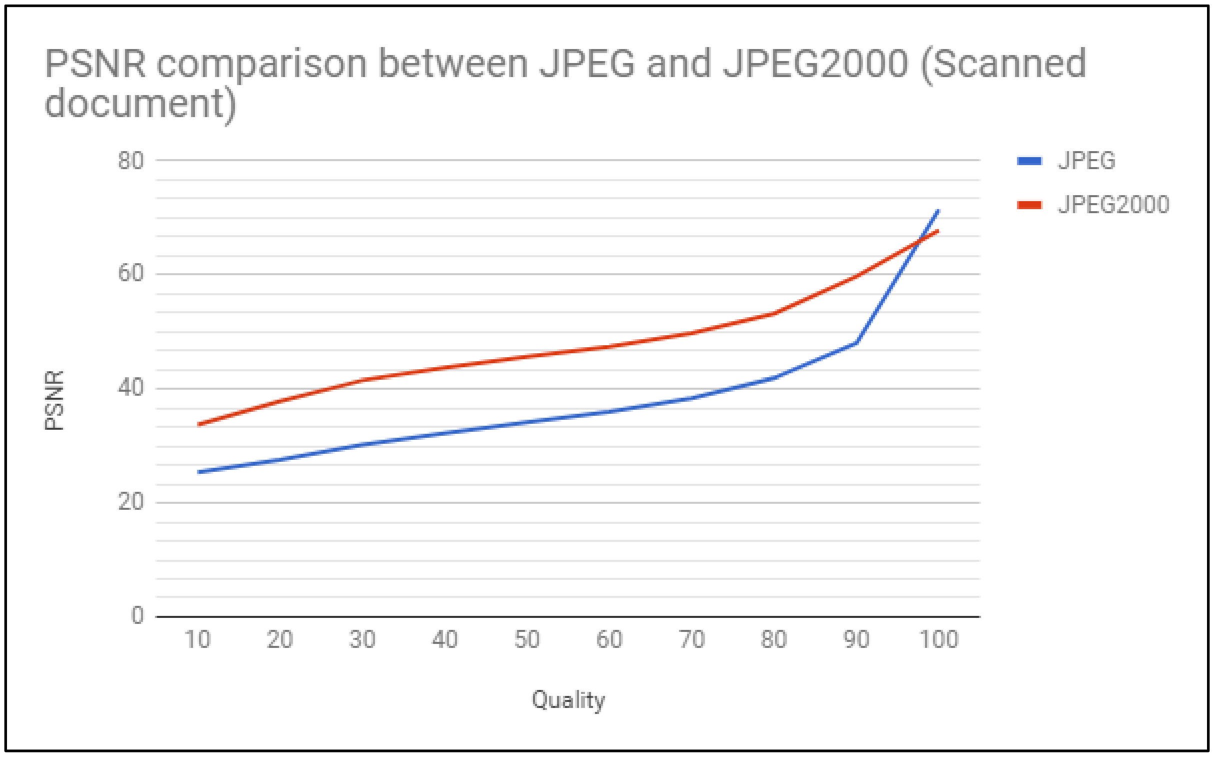
Compression ratio calculated from the findings in JPEG	Quality parameter that was passed in JPEG to achieve that ratio	PSNR of image compressed using the JPEG2000 by passing the compression ratio
76.548	10	33.7241
59.2612	20	37.8504
49.1434	30	41.5184
43.6481	40	43.6908
39.6647	50	45.5867
36.3553	60	47.333
32.7417	70	49.768
28.7192	80	53.1517
22.8977	90	59.6751
12.2742	100	67.7596

Compressing the original image with the lossless algorithms, gave me the following results:

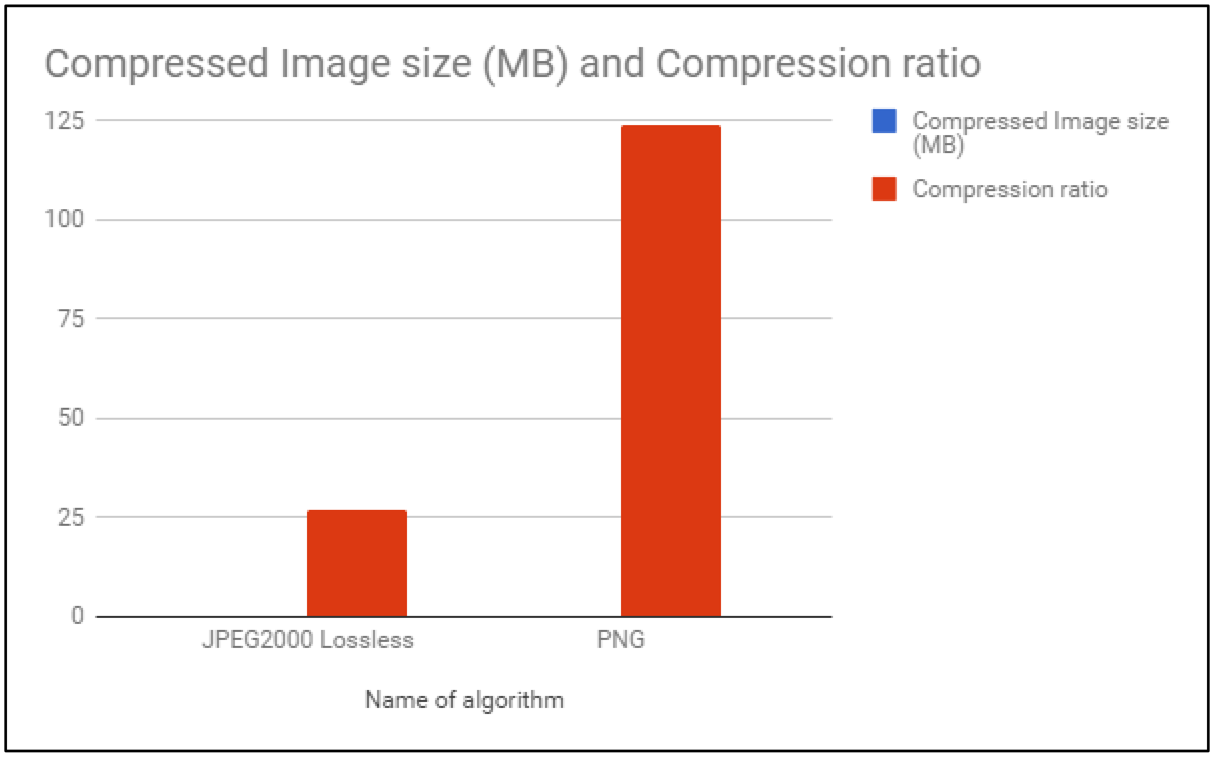
Algorithm's Name	Compressed image size (Megabytes)	Compression ratio (original image's size/compressed image size in megabytes) [1]

JPEG2000 lossless	0.41029	27.0101
PNG	0.089328	124.0596

Graphical representation of the data found/calculated for lossy algorithms:



Graphical representation of data found/calculated for lossless algorithms:



5. Conclusion

Based on the results from my primary research, I could understand how the algorithms performed with different images and therefore conclude which algorithm has performed the best.

High definition image:

The lossy image compression algorithms, JPEG and JPEG2000, performed very well in terms of compression that they were able to achieve. The figure xx clearly shows that JPEG2000 was able to perform better than JPEG in each quality parameter. When the quality parameter was set to 100%, image compressed using JPEG had PSNR value of 42.0475 whereas, the image compressed using JPEG2000 had PSNR value of 48.022. As the compression ratio was same for both the algorithms, it can be said that JPEG2000 is actually a better and an efficient successor of

JPEG when it comes to high definition images. Coming to lossless image compression algorithms, there was not much of a difference in the findings between the two algorithms (PNG and JPEG2000 Lossless). Therefore, it is difficult for me to conclude which algorithm performed better.

Grayscale image:

The findings for this image in terms of lossy compression algorithms were quite interesting. JPEG2000 was performing better than JPEG until the quality parameter was set to 100%. JPEG outperformed JPEG2000 at this point. The difference between the PSNR values of JPEG and JPEG2000 was 5.0963. Exploring why this happened is out of the scope of this essay. But, JPEG performed the best in grayscale image compression under the lossy variation. JPEG2000 lossless performed exceptionally well compared to PNG in the lossless variation. The size of the image, compressed using PNG was approximately twice in size when compared to JPEG2000 lossless.

Scanned document:

One thing that caught my attention was the high PSNR values in lossy compression. The highest PSNR value that was achieved among all the compressions was 58.4847 (grayscale image compression using JPEG at 100% quality) but in this case, i.e Scanned document, the PSNR went to 71.4491 (100% quality JPEG). Once again, JPEG2000 failed to compete with JPEG when the quality was set to 100%. JPEG’s PSNR value at 100% quality was 71.4491 where as JPEG2000’s PSNR value was 67.7596. Except this case, JPEG2000 performed better than JPEG in other quality parameters. The results from lossless image compression were unbelievable. PNG was able to compress an image of 11.08 MB to 0.089 MB, that is a compression ratio of nearly 124.06. The lossy image compression algorithms were not able to achieve this even at 10% quality, but PNG was able to it without compromising with the quality of the image.

From the above observations, I can conclude that JPEG2000 is better than JPEG in most of the cases. But, under certain circumstances, JPEG2000 was not able to perform better than JPEG. I believe that JPEG200 still has some space for improvisation. Coming to the lossless algorithms, PNG is still the most efficient algorithm when it comes to scanned documents. But, in the other two type of images, JPEG2000's lossless version performed better than PNG.

6. Limitations of the essay

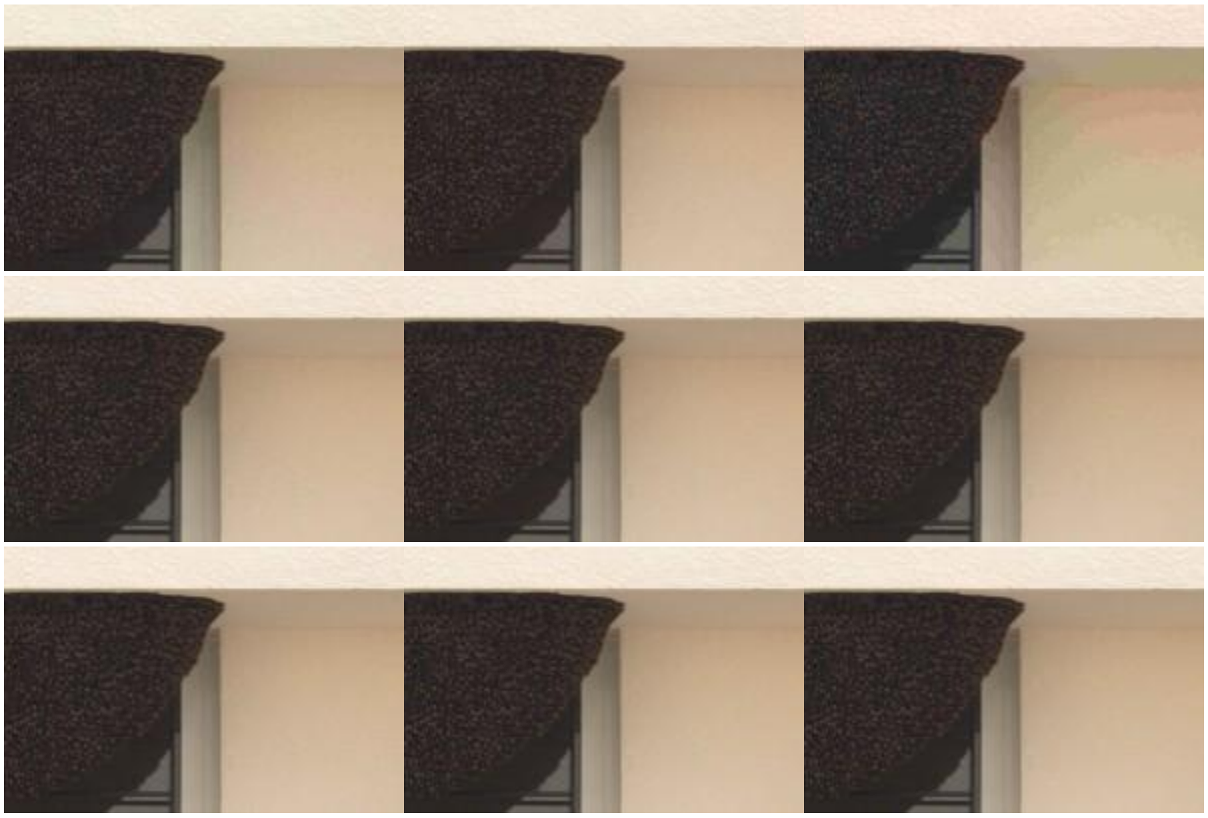
This essay's scope is limited to just four compression algorithms that too with just 3 types of images. There are many different types of images and algorithms, which were not discussed in this essay. The other limitation of this essay is that, this essay's main purpose was to decide whether JPEG2000 is a better compression algorithm than the other two algorithms. But in certain cases where JPEG2000 was not able to perform better than the other algorithms, I could not explore it further to understand the cause of underperformance.

7. Appendix

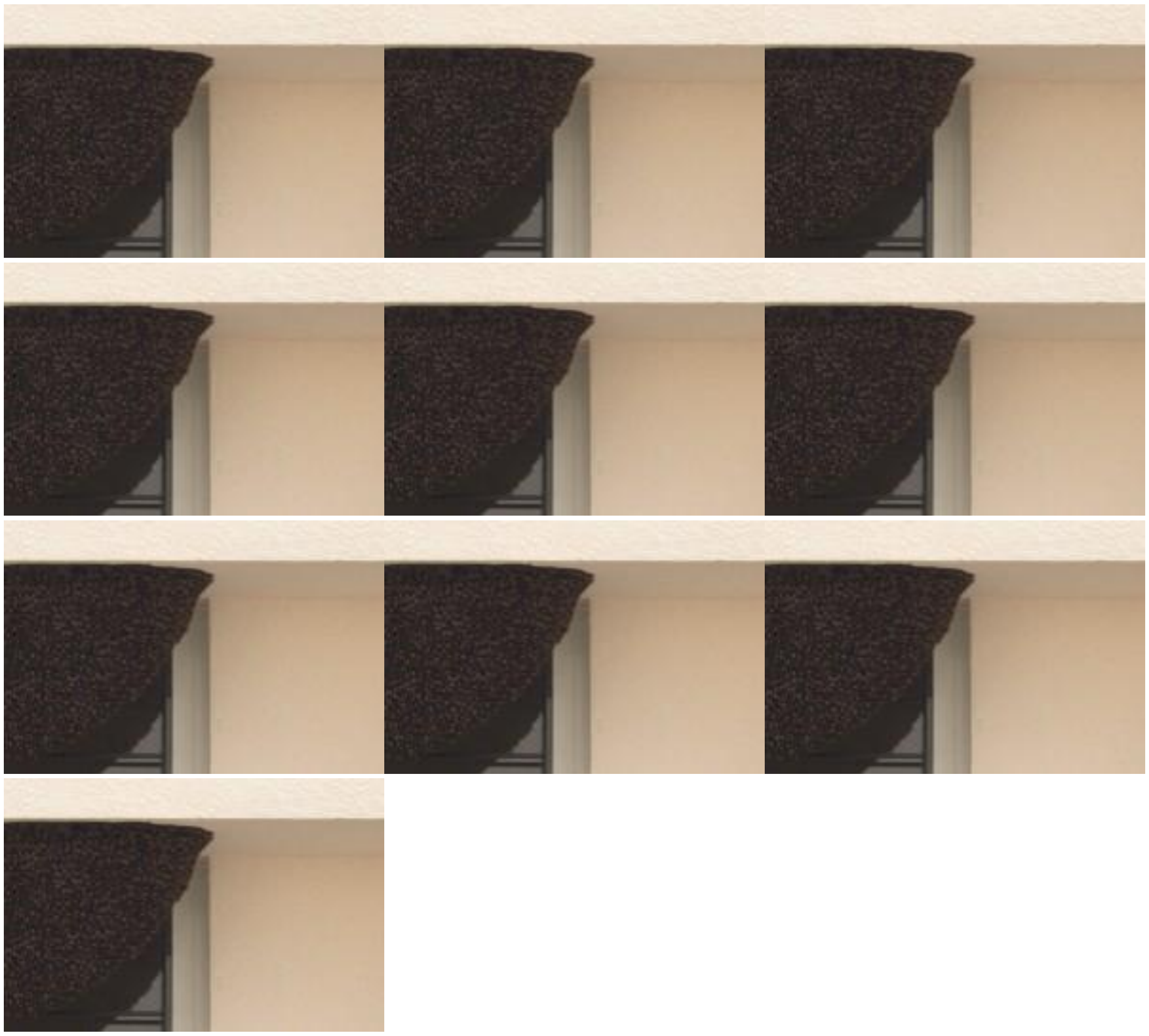
7.1. Compressed images

7.1.1. High resolution images

JPEG



JPEG2000

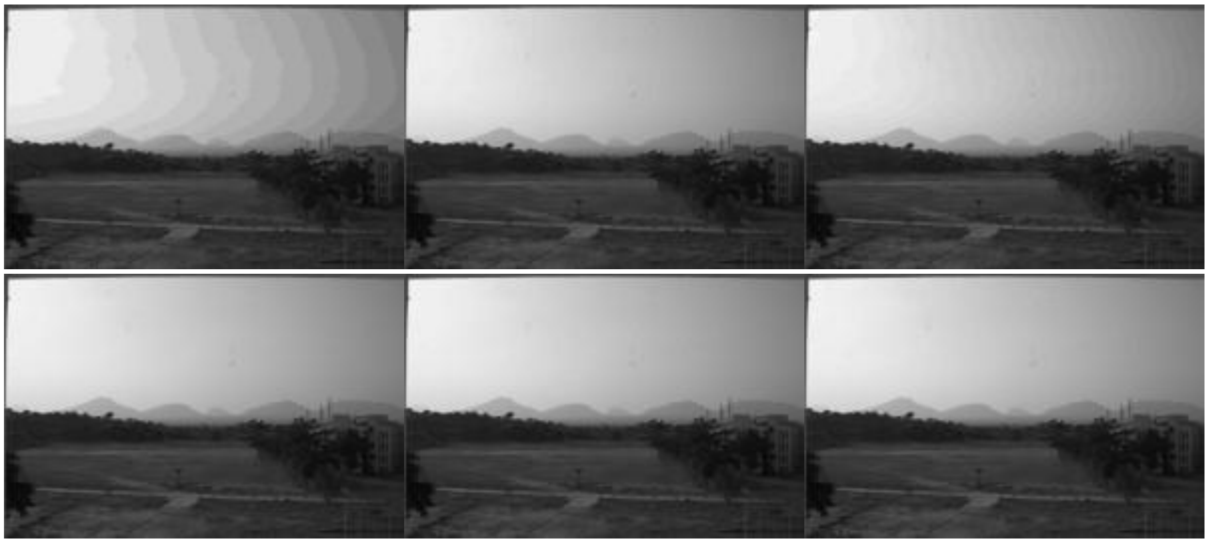


PNG

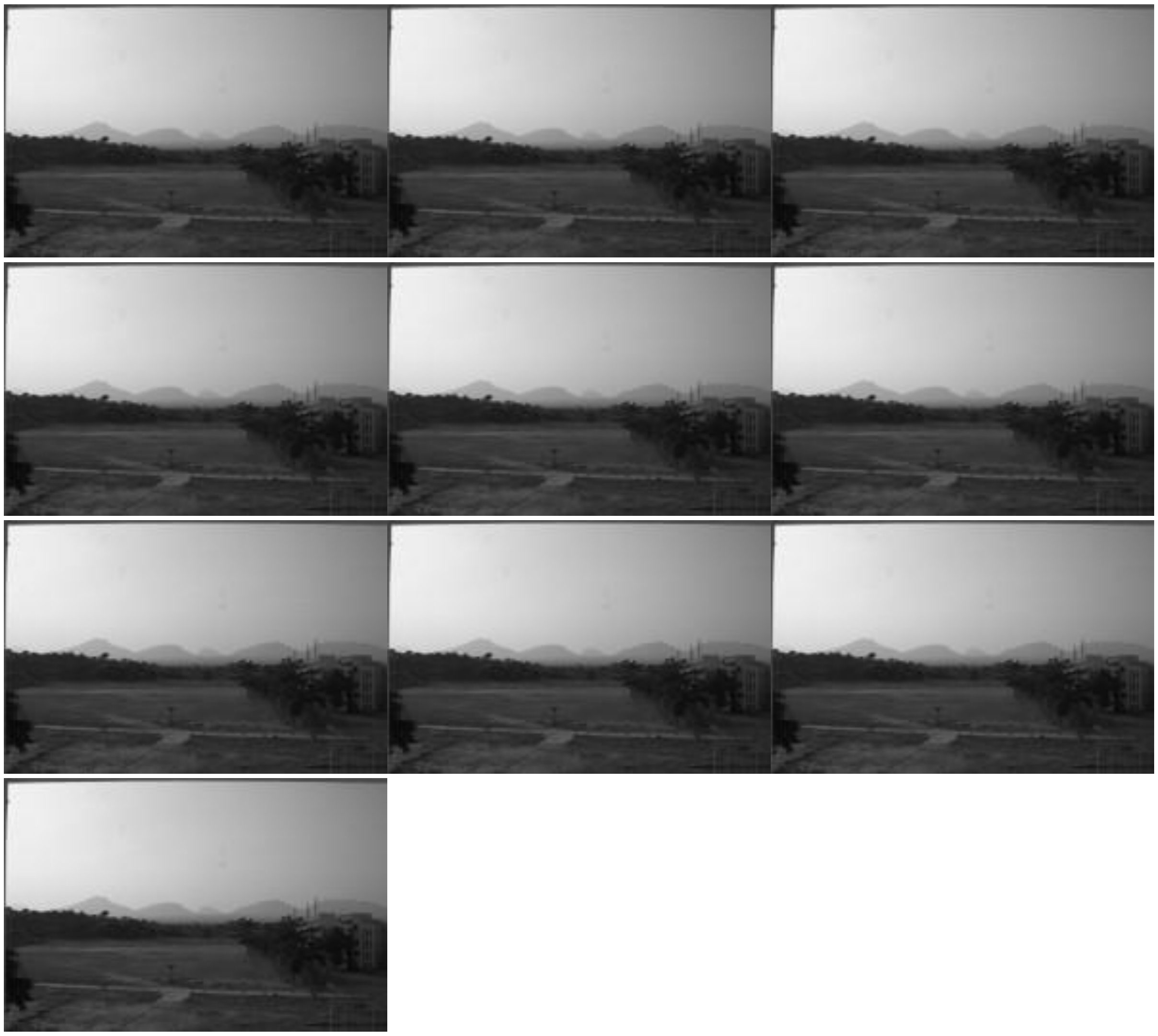
JPEG2000 lossless

7.1.2. Grayscale images

JPEG



JPEG2000

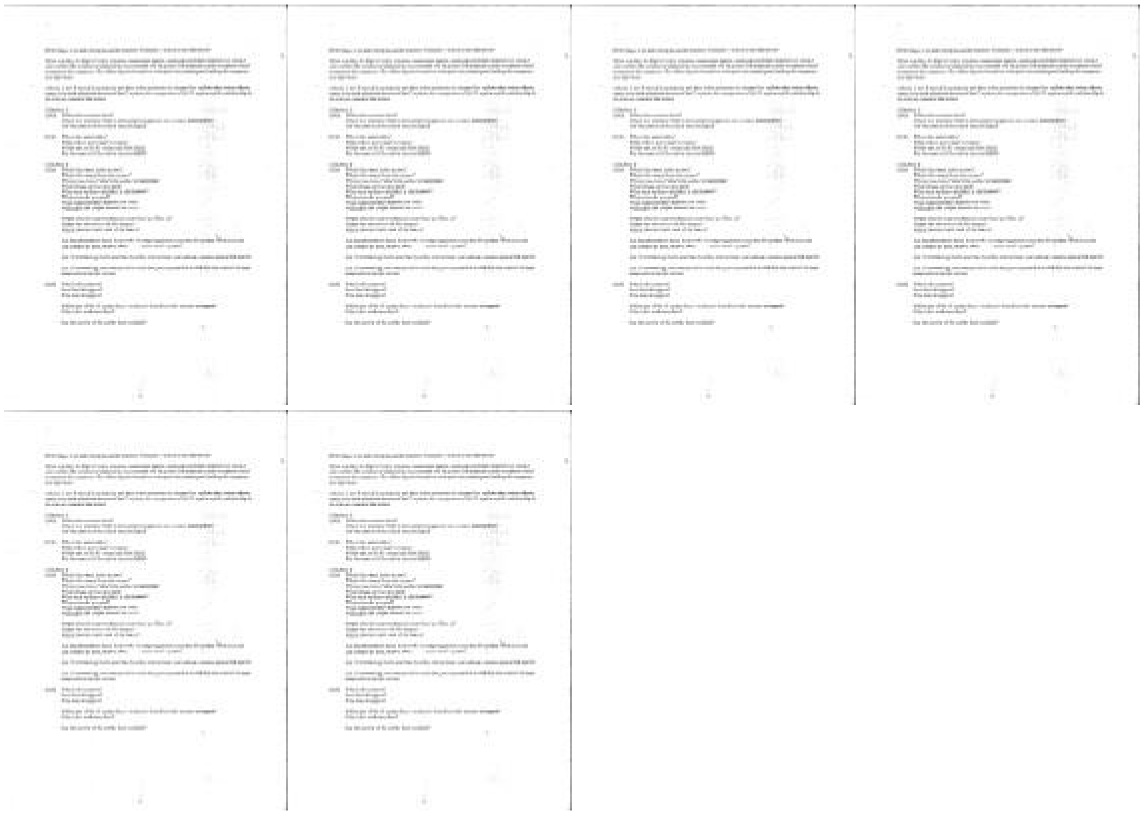


PNG

JPEG2000 lossless

7.1.3. Scanned images

JPEG



JPEG2000

PNG

JPEG2000 Lossless

For the sake of attaching the image into the essay, I have resized the image and put into the appendix. Because of the resizing, the dimensions and the file size have changed. The original files can still be accessed from my google drive link:

https://drive.google.com/open?id=1OhqWLN5WmUqH-3YJqxBaxC4rhSbxm63u

For example if you wish to access the grayscale image which was compressed using JPEG at 10% then, click on grayscale images then select JPEG after that click on the appropriate file.

All the folders are sorted in the same way for easy access.

7.2. Programs

I have written the programs in Matlab. I wrote three programs, which were more or less the same for every image. Only for the JPEG2000, as there were unique compression ratio every time, I wrote different programs for each image.

As all the programs were more or less the same, I'll mention one example of each:

1. BMP to JPEG.
2. BMP to JPEG2000.
3. BMP to PNG and JPEG2000 Lossless

BMP to JPEG:

```
%Original BMP File read using imread function.

bmp=imread('S:\IB\Extended Essay\Images\grayscale_image.bmp');

%Original BMP file's information read using dir function.

original_file=dir('S:\IB\Extended Essay\Images\grayscale_image.bmp');

disp(original_file.bytes/(1024*1024))

name='grayscale_';

disp("JPEG");

x=0;

while x<100

    x=x+10;

    percentage=strcat(name,num2str(x));

    ext=strcat(percentage,'.jpg');

    disp(ext);

    imwrite(bmp,ext,'jpg','quality',x);

    s=dir(ext);

    current_image=imread(ext);

    disp(["Image Size:" s.bytes/(1024*1024)]);

    disp(["Compression ratio:" original_file.bytes/s.bytes]);

    disp(["PSNR: " psnr(current_image,bmp)]);

end
```

BMP to JPEG2000:

```
imwrite(bmp,'grayscale_jpeg2000_10.jp2','jp2','mode','lossy','CompressionRatio',170.4393);
imwrite(bmp,'grayscale_jpeg2000_20.jp2','jp2','mode','lossy','CompressionRatio',145.1801);

imwrite(bmp,'grayscale_jpeg2000_30.jp2','jp2','mode','lossy','CompressionRatio',124.9006);

imwrite(bmp,'grayscale_jpeg2000_40.jp2','jp2','mode','lossy','CompressionRatio',110.4713);

imwrite(bmp,'grayscale_jpeg2000_50.jp2','jp2','mode','lossy','CompressionRatio',98.3032);

imwrite(bmp,'grayscale_jpeg2000_60.jp2','jp2','mode','lossy','CompressionRatio',87.1778);

imwrite(bmp,'grayscale_jpeg2000_70.jp2','jp2','mode','lossy','CompressionRatio',72.1622);

imwrite(bmp,'grayscale_jpeg2000_80.jp2','jp2','mode','lossy','CompressionRatio',53.4456);

imwrite(bmp,'grayscale_jpeg2000_90.jp2','jp2','mode','lossy','CompressionRatio',29.6221);

imwrite(bmp,'grayscale_jpeg2000_100.jp2','jp2','mode','lossy','CompressionRatio',7.2986);
```

BMP to JPEG2000 lossless and PNG:

```
%JPEG2000

imwrite(bmp,'grayscale_jpeg2000_loseless.jp2','jp2','Mode','lossless');

ext='grayscale_jpeg2000_loseless.jp2';

s=dir(ext);

disp(["Image Size:" s.bytes/(1024*1024)]);

disp(["Compression ratio:" original_file.bytes/s.bytes])

current_image=imread(ext);

disp(["PSNR:" psnr(current_image,bmp)]);



%PNG

imwrite(bmp,'grayscale_png.png','png');

ext='grayscale_png.png';

s=dir(ext);
```

```
disp(["Image Size:" s.bytes/(1024*1024)]);

disp(["Compression ratio:" original_file.bytes/s.bytes])

current_image=imread(ext);

disp(["PSNR:" psnr(current_image,bmp)]);
```

8. Bibliography

"Dell's UP3218K 8K monitor is stunningly ahead of its time - The Verge." 12 Sep. 2017, https://www.theverge.com/2017/9/12/16293326/dell-8k-monitor-review-ultrasharp-up3218k. Accessed 10 Nov. 2017.

"All About Digital Photos - What Is a Digital Photo - Rideau-Info." http://www.rideau-info.com/photos/whatis.html. Accessed 12 Nov. 2017.

"MPEG-4, Advanced Video Coding (Part 10) (H.264)." 21 Feb. 2017, https://www.loc.gov/preservation/digital/formats/fdd/fdd000081.shtml. Accessed 12 Nov. 2017.

"Reducing Your Website's Bandwidth Usage - Coding Horror." 5 Mar. 2007, https://blog.codinghorror.com/reducing-your-websites-bandwidth-usage/. Accessed 13 Nov. 2017.

"Everything You Need to Know About Image Compression - noupe." 3 Feb. 2010, https://www.noupe.com/design/everything-you-need-to-know-about-image-compression.html. Accessed 29 Nov. 2017.

"Chapter 2 Digital Image Compression." http://digital.cs.usu.edu/~xqi/Proposal/Chapter2.pdf. Accessed 29 Nov. 2017.

"4. Quantization and Data Compression - Purdue Engineering." https://engineering.purdue.edu/~ipollak/ece302/SPRING12/notes/25_Quantization_and_Compression.pdf. Accessed 1 Dec. 2017.

"Chapter 2 Digital Image Compression." http://digital.cs.usu.edu/~xqi/Proposal/Chapter2.pdf. Accessed 2 Dec. 2017.

"Data Compression with Run-length Encoding - stoimen's web log." 9 Jan. 2012, http://www.stoimen.com/blog/2012/01/09/computer-algorithms-data-compression-with-run-length-encoding/. Accessed 2 Dec. 2017.

"Greedy Algorithms | Set 3 (Huffman Coding) - GeeksforGeeks." http://www.geeksforgeeks.org/greedy-algorithms-set-3-huffman-coding/. Accessed 3 Dec. 2017.

"LZW (Lempel–Ziv–Welch) Compression technique - GeeksforGeeks." http://www.geeksforgeeks.org/lzw-lempel-ziv-welch-compression-technique/. Accessed 4 Dec. 2017.

"BMP (Bitmap) Definition - The Tech Terms Computer Dictionary." https://techterms.com/definition/bmp. Accessed 4 Dec. 2017.

"PNG (Portable Network Graphics) Home Site - libpng.org." 4 Nov. 2017, http://www.libpng.org/pub/png/. Accessed 5 Dec. 2017.

"How does JPEG actually work? – Developing for Developers." 12 Apr. 2006, https://blogs.msdn.microsoft.com/devdev/2006/04/12/how-does-jpeg-actually-work/. Accessed 5 Dec. 2017.

"Jpeg Image Compression Using Discrete Cosine Transform - A ... - arXiv." https://arxiv.org/pdf/1405.6147. Accessed 5 Dec. 2017.

"Lossy Data Compression: JPEG - CS Stanford." https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/coeff.htm. Accessed 5 Dec. 2017.

"Write image to graphics file - MATLAB ...." https://www.mathworks.com/help/matlab/ref/imwrite.html. Accessed 10 Dec. 2017.

"How to calculate image compression ratio? - MATLAB Answers ...." 10 Aug. 2015, https://uk.mathworks.com/matlabcentral/answers/237620-how-to-calculate-image-compression-ratio. Accessed 15 Dec. 2017.

"Peak Signal-to-Noise Ratio (PSNR) - MATLAB psnr - MathWorks." https://www.mathworks.com/help/images/ref/psnr.html. Accessed 15 Dec. 2017.

"Peak Signal-to-Noise Ratio as an Image Quality Metric - National ...." 11 Sep. 2013, http://www.ni.com/white-paper/13306/en/. Accessed 15 Dec. 2017.

A.H.M. Jaffar Iqbal Barbhuiya, Tahera, K. Hemachandran. "An Approach for Color Image Compression of JPEG and PNG Images using DCT and DWT." *IEEE Computer Society* (2014): 129-133. Document.

Aguilera, Paula. "Comparison of different image compression formats." Project Report. n.d. Document.

Cruz, Diego Santa, Touradj Ebrahimi and Charlios Christopoulos. "The JPEG 2000 Image Coding Standard." *Dr. Dobb's Journal* (Apr 2001): 46. Document.

Dipert, Brian. "Compression - put images on a diet." *EDN* (Jun 18, 1983): 71-86. Document.

Rodrigues, Lins and. "Assessing File Formats for Static Image Transmission." *ITS* (2006): 310-314. Document.

S, Anitha. "Comparison of Compression Technigues." *International Journal of Advanced Research in Computer Sciences* (Jan-Feb 2014): Volume 5, No.1. Document.

Savetz, Kevin. "The care and feeding of PNG." *ProQuest* (Jun 2001): 38. Document.

Wooldridge, Mike. "Low-Fat Graphics for the Web." *MacWorld* (Jan 2000): 111-112. Document.