IMAGE COMPRESSION BASED ON NON-PARAMETRIC SAMPLING IN NOISY ENVIRONMENTS

Kishan Y Narotam (717 931)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: This paper presents the research, design and implementation of an image compression scheme involving: creating holes in an image; encoding and transmitting the image through a simulated noisy channel; and filling these holes and reconstructing the image. The algorithm to create and fill the holes used existing techniques such as DCT and FIC as the foundational idea. The compression scheme was implemented in *MATLAB* and tested using multiple images of different types (pattern, landscape and high contrast). The implemented system was able to give an average of compression ratio of 1.2543, 1.6753, 2.1102 for all pattern, landscape and high contrast images, respectively. The scheme is also able to outperform DCT compression techniques as more noise is introduced to the channel by yielding PSNR values $2dB$ higher. The created compression scheme is tested with known images such as mandrill, cameraman and peppers against DWT, however is not as effective in terms of compression ratio, but yields better results as errors are introduced. The implemented system is critically analysed with recommendations of parallel programming and the use of neural networks given.

Key words: compression ratio, DCT, holes in image, image compression, MATLAB, PSNR

1. INTRODUCTION

The world is in the Fourth Industrial Revolution, and the development, usage and demand of multimedia products is continuously growing. With the resolution of images and videos reaching higher levels of quality and clarity, storage and network bandwidth are beginning to become insufficient [1]. Image compression plays an important role in the digital environment, with many techniques already existing, namely various lossy and lossless schemes being utilized currently [2].

This paper presents the research, design and implementation of an image compression technique that involves: creating holes in the image; encoding and transmitting the image in a simulated noisy channel; filling in the holes; and finally reconstructing the image. The paper is structured as follows: Section 2. discusses the project background including the project specification, a literature review and the objectives of the project. Section 3. looks at the overview of the implemented system, whereas Section 4. provides an in-depth breakdown thereof. Section 5. reviews the testing of the project and the subsequent results. An analysis of the implemented system as well as future recommendations are given in Section 6., and lastly Section 7. presents the concluding arguments.

2. PROJECT BACKGROUND

Image compression is a form of data compression with the goal of reducing the amount of storage that is required to store the original image. Compression of the image can be seen by reducing the amount of redundant data within the image, while the compressed image must resemble the original image as much as possible [2].

2.1 Project Specification

The objective of the project is to create a robust scheme of image compression that must involve: creating holes in an image; encoding and transmitting the image; simulating the channel for transmission that introduces errors; and filling the holes appropriately and reconstructing the image. The implementation of the project can be accomplished in software utilizing the *MATLAB* environment, and has three major constraints, namely: ethics, budget and time. The project requires ethical clearance from the university; to be completed within the allocated budget $R\ 1,200$; and to be completed within a six week time frame. Appendix G presents all information relating to project management, including minutes recorded for all meetings.

2.2 Literature Review

There are multiple methods and techniques that currently exist for image compression that can be categorized in either lossy or lossless compression. A lossy compression scheme sees that some information from the original image is lost in the final compressed image. A lossless compression scheme sees that all information in the original image is retained within the final decompressed image, noting that the compression is seen by the encoding of the information in the image [3].

2.2.1 Creating and Filling Holes Yoon and Chung [4] discussed texture image extrapolation for compression due to the repetitive nature of the texture image. The first approach explored, utilizes non-parametric sampling by pattern matching, whereby data was transmitted, such that the quality of the image would not be worsened. A texture image is tested by being transmitted with a hole, and the hole is filled using pattern matching at the receiver [4], with no information given on how the hole is created within the image. Given a specific pixel which is required to be filled, the entire image is searched for an area which closely resembles where the hole/empty space is to be filled [4]. The alternative used for the pattern matching assumed the Markov model on the image, where one large hole is found within the image. The final case considered multiple holes across the image, assuming a sub-sampled image is known and used. The sub-sampled image is then used to estimate empty pixels by matching blocks, with new estimated pixels being incorporated into the estimation. The utilization of the sub-sampled image proved to yield satisfactory results [4].

2.2.2 Discrete Cosine Transform The Discrete Cosine Transform (DCT) is the most common use of lossy compression utilized in the JPEG process. The process is done by breaking the image down into separate 8×8 blocks of pixels and applying the DCT [5]. The DCT separates the image into different frequencies, as the equation representing DCT is the *real* part of the Fourier Transform, as seen in Equation 1 in Appendix F1. The compression in this technique is seen during the quantization part of the process where frequencies that hold no importance in the image are removed, ultimately removing part of the original data from the image [5]. The final reconstructed image will display a sense of distortion, however this can be controlled and adjusted accordingly to affect the overall level of quality and compression of the image [5].

2.2.3 Run Length Encoding Run Length Encoding is a form of lossless image compression that reduces the data into small sequences with the benefit of maintaining the quality of the image. The compression is seen where data is repeated, such as a colour being repeated for multiple adjacent pixels. However, compression is not always guaranteed, as in some cases where consecutive adjacent pixels have different values, the compression scheme performs poorly [6].

3. SYSTEM OVERVIEW

The overall system block diagram of the implemented image compression technique is presented in Figure 1. The green blocks represent all processes that involve displaying and storing the image; the red blocks indicate the creation and filling of the respective holes in the image; the blue block presents the DCT technique that was chosen to add a higher level of compression and complexity to the scheme; and the orange blocks indicate the channel of the image, from encoding, introducing errors and decoding the image. In the figure, three different compression techniques are displayed in the entire block diagram, these being: creating holes in the image, passing it through the channel, filling the holes and displaying the compressed image; performing DCT compression on the image, passing it through the channel and displaying the compressed image; and combining the aforementioned two techniques of creating holes, performing DCT, passing the image through the channel, filling the holes and displaying the compressed image. All three techniques will use the first three steps of the system, before undergoing any one of the three compression schemes developed.

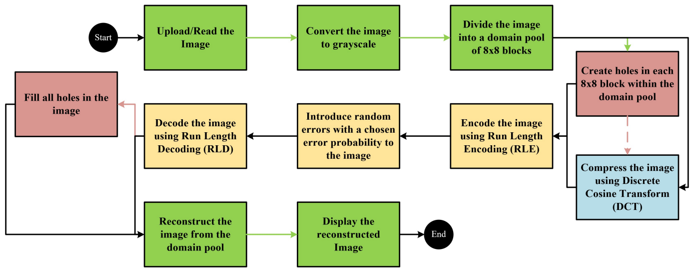


Figure 1: Block diagram of the implemented system

4. SYSTEM IMPLEMENTATION

4.1 Uploading/Reading the Image

Any chosen image is read into the *MATLAB* workspace using the `imread()` function. After being read into the workspace, any image of colour is seen to have three dimensions: the height of the image; the width of the image; and the colour map, all values being 8-bit unsigned integers.

4.2 Converting the Image to Grayscale

Since the colour map is present in the read image, it is removed, as grayscale images measure light intensity only. Considering that the colour image has all values as 8-bit unsigned integers, removing the colour map leaves all pixels in a possibility of 256 shades of gray ranging from 0 to 255 [2]. The `rgb2gray()` function averages out the red, green and blue components of the colour map in the image. This results in an image containing only two dimensions: the height and width of the image, with all values being 8-bit unsigned integers. The grayscale image is considered and referred to as the *uncompressed/original* image from this moment on.

4.3 Dividing the Image into the Domain Pool of 8×8 blocks

The image is divided into smaller 8×8 pixel blocks as this is the basis of modern compression schemes such as DCT and block based fractal image compression [5, 7]. The assumption is made that all images used in this project have dimensions divisible by 8 allowing for the domain pool to be created without errors. A `cell` array is created of 8×8 pixel blocks. The size of the array is defined as the $numberOfBlocksAcross \times numberOfBlocksDown$. Knowing the total number of 8×8 blocks in the image, the image is reduced into the respective blocks being inserted into the array from left to right, top to bottom. This means that the top left block will be indexed as 1, and the block directly to the right of it as 2, and so on. Equation 2 in Appendix F1 shows how to determine the x or y coordinate of the top left pixel of the desired indexed block. Having divided and broken down the image into the domain pool, each block in the array is an 8×8 pixel block.

4.4 Creating Holes in each Block within the Domain Pool

The foundational idea on where and how to create the holes came from A. Selim et al's block based fractal image compression [7]. A reference block is chosen within each respective block in the domain pool, which is the center pixel within the block [7]. That same idea is brought forward in creating holes in the image. A hole in this implementation is defined as a square area of either 2×2, 4×4 or 6×6 pixels with each pixel in the square area having a value of 0.

Figure 76 in Appendix E shows how each 8×8 pixel block is referenced and coordinated within the array. Holes are created by starting in the center 2×2 (blue) square within the 8×8 block. The average value of the blue square is calculated using the `mean()` function. The average value of the 2×2 square is now compared to each pixel within it by calculating the Chebyshev distance between the average pixel value and each pixel within the blue square. If all distances between the average and each pixel is less than 6, a hole can be created in the 2×2 sqaure. However, before the hole is created, the process of calculating the average and checking the Chebyshev distance is done for the respective 4×4 and 6×6 squares, ultimately determining the size of the hole that can be created in the image. The largest hole size that can be created is a 6×6 pixel block, as the surrounding pixels are used as reference blocks in order to fill these holes. Algorithm 1 in Appendix F3 gives a high-level version of the aforementioned method to create holes.

4.4.1 Chebyshev Distance Equation 3 in Appendix F1 gives the mathematical formula of the Chebyshev distance [8]. In *MATLAB*, when the `pdist()` function is used, and the distance metric chosen is the Chebyshev distance, it is regarded as the maximum coordinate difference between two vectors. Ten simulations were run, to determine what distance value is chosen for the implemented system and the results can be found in Appendix E1. The distance values ranged from 1 to 10, and it was seen that a value of 6 allows the image to create a relevant number of holes, and in reconstruction, gave the best looking final image.

4.5 DCT Compression

MATLAB has multiple ways of performing the DCT transformation with built-in functions, however since the images used have been reduced to two dimensional matrices, the `dct2()` and `idct2()` functions are used to perform a 2-D discrete cosine transform and 2-D inverse discrete cosine transform respectively. The first step in the implemented DCT compression calculates and displays the frequency intensity of the original image. The actual compression of the DCT is done for each block in the domain pool. First the `dct2()` function is done on the image (with or without holes, depending on the chosen method of compression). With the transform is applied, the quantized image is now known. The user can be prompted to select a compression depth value between 1 and 8 inclusive. The compression depth value dictates the amount of data that will be removed from the image. If the compression depth is chosen to be 7 or 8, the level of compression seen is low as the quality of the image is maintained. Consequently, if a compression depth of 1 or 2 is chosen, the compression level seen is significantly higher, however the trade-off is that the overall quality of the image is drastically reduced. Since the transformation of the 8×8 blocks alter the values of each pixel, a compression depth of 7 is chosen for the combined implementation. The reason for this is that the holes can still be filled in the final reconstruction of the image.

4.6 Run Length Encoding and Decoding

The Run Length Encoding and Decoding utilized in the project was adapted from Abdulrahman Siddiq's implementation of the encoding scheme. The license file can be seen in Listing 1 in Appendix F2. The encoding works on row vectors, and so each block in the domain pool is analysed row by row. The encoding begins by seeing the first pixel value in the selected row, and giving that pixel value a count of 1. The second element in the row is checked against the previous element, if the values are the same, the count for that specific pixel value is increased by one. This process is repeated for each value in the row, for each row within each block in the domain pool. As a result of this, two sets of data are transmitted through the channel: the pixel values and their respective counters.

The decoding of each row works by checking the respective counters of each specific value. Depending on how many times a specific pixel value occurs according to its respective counter, the decoding algorithm will reconstruct each row within every 8×8 block within the domain pool.

4.7 Introducing Errors

The channel is simulated by the introduction of errors to the image. The probability of errors being introduced to the image is based on a "coin-flip" scenario, where a random number is generated between 1 and 1000 using the `rand()` function. The user is able to enter and control the probability of an error occurring, through a created input dialogue block. In conjunction to this, the user is able to introduce errors in one of two methods: bit flip or ones compliment. The error is subject to affect only the encoded pixel value and not the pixel count, as this is an assumption made in the project, as the errors equate to noise being present in the channel.

4.7.1 Bit Flip Error Introduction The bit flip error introduction affects only one bit of the transmitted pixel value. First, the pixel value is converted from decimal into a binary vector, using the `decimalToBinaryVector()` function. Only the necessary number of bits are converted as opposed to the available 8-bits. This means that if a pixel has a value of 10 for example, only 4-bits are required to represent this value. This conversion of the necessary bits is where the compression is seen in the implemented technique. Following this, depending on the number of bits required to represent a value and the probability of an error being introduced, either the most significant bit or the least significant bit or anything in-between can be affected.

4.7.2 Ones Compliment Error Introduction The ones compliment error introduction affects all bits required to represent a specific pixel value. Performing a ones compliment or inversion of the value, introduces more noticeable noise to the image. For example, should a pixel have the value of 124, which requires 7-bits to be represented, then have an ones compliment error introduced to it, the pixel now has a value of 3. Although the bit flip error introduction is more likely to be seen in modern channels, the ones compliment error introduction was introduced to determine whether the final image reconstructed is recognisable and to determine how noticeable the noise is depending on the error probability.

4.8 Filling the Holes

Since non-parametric sampling is used, once decoded the receiver has no indication of where any holes or errors have been introduced. However, it stands to reason that the inverse methodology of creating holes can be used to detect where holes are, this being a novel idea. As stated in Section 4.4 and in Figure 76, the average of the 2×2 square is calculated, and the Chebyshev distance is checked between the average and each respective pixel in the blue square. If the distance is less than 6, there is a hole present, and so the process is repeated, checking to see if a hole is of size 4×4 or 6×6. Unlike searching other parts of the image to see if the surrounding pixels are similar, the holes are filled using a smoother transition. Starting at the top left pixel of the hole, the average pixel value of the pixels directly above, directly to the left and directly above-left is calculated. This average value is used to fill the top left pixel of the hole. This idea of filling the hole, pixel by pixel, moving left to right, top to bottom, gives the image a smoother transition of colour within reason. Algorithm 2 in Appendix F3 presents a high-level algorithm on how the filling of the holes is implemented.

4.9 Reconstructing the Image

Throughout the process of the compression scheme, the image is compressed through the smaller blocks that create the domain pool. Reconstructing the image requires the `cell` array to be combined together to form the final compressed image. Variables storing the number of 8×8 blocks across and down the image are stored, and are utilized to reconstruct the image, starting at the top left, moving left to right, top to bottom.

5. TESTING AND RESULTS

A variety of test cases were presented and trialled against the implemented system and the various compression schemes. The tests ranged from keeping the noise or error introduction to a minimum in order to test the created algorithms, to increasing the probability of error's being introduced and determining the mean squared error and peak signal-to-noise ratio. The compression ratios were determined using Equation 6 in Appendix F1. All results, figures and tables have been presented in Appendix D.

Three different types of images were tested: landscape images, pattern/texture images and high contrast images. These images are chosen for the same reason known images such as the *cameraman*, *mandrill* and *peppers* were used. Landscape images are chosen as they have a combination of some repetitive colour data as well as many edges and features. Pattern/Texture images are chosen due to the repetitive nature of the images. The high contrast images were chosen for the combination of repetitive nature as well as details within the image. All images were taken from `Unsplash.com`, which provide free-to-use images and the respective photographers are acknowledged at the end of the report. The largest image size chosen is approximately 2000×1200 pixels. The chosen images can be found in Appendix E2. The probability of error being introduced varies for each test case, as it simulated a worse-case scenario for modern channels. For example, according to the IEEE 802.15.4 standard, a receiver's sensitivty must yield a packet error rate of less than 1%, whereas the maximum allowed error vector magnitude allowed is 35% [9].

5.1 *Test Case 1: Testing the Same Image at Different Sizes*

The purpose of this test case was to see how the same image at different resolutions would perform in terms of compression ratio using the *Holes*-only scheme. The probability of error introduced was 10%.

Table 2 presents a table of the compression ratio, PSNR and MSE of the pattern images at different sizes along with Figure 2 in Appendix D1.1 showing a graph of the compression ratio. It was seen that the average compression ratio across all images was 1.303765. The pattern images show that at 40% of the original size, there is a point of inflection, where the compression ratio increases as the size of the image decreases. This is valid, as when a pattern image is reduced in resolution, the pixel colours are repeated more times than at a higher resolution, as adjacent pixels would have differences of 1. As mentioned in Section 2.2.3, the RLE scheme performs poorly in this scenario. Subsequently, as the size of the image increases, it seems that the noise becomes more negligible to the naked eye.

Table 3 presents a table of the compression ratio, PSNR and MSE of the pattern images at different sizes along with Figure 23 in Appendix D1.2 showing a graph of the compression ratio. In the case of landscape images, with the exception of the Simon Matzinger photo, the compression ratio remained fairly constant as the size changed, with an average compression ratio of 1.647453. Regardless of this, it shows that as the size of the image increases, so does the compression ratio. The Simon Matzinger photo was chosen due to its smooth colour transition, having similar properties as a pattern image. Like the pattern images, noise begins to become minor to the naked eye as the image size increases.

Table 4 presents a table of the compression ratio, PSNR and MSE of the pattern images at different sizes along with Figure 39 in Appendix D1.3 showing a graph of the compression ratio. The average compression ratio for high contrast images was calculated as 2.09093. The high contrast images show that as the resolution of the image increases, so does the compression ratio. This is due to the data in high contrast images having significant repetition, proving that the encoding scheme is fit for these types of images.

5.2 *Test Case 2: Average Compression Ratios of the Different Image Types*

The purpose of this test case was to determine the average compression ratio of the different image types at a high definition resolution of approximately 1200×800 pixels. No errors are introduced to these tests as a way of proving whether the designed compression system of *Holes*-only works. Appendix D2 presents all the tables, graphs and images tested for this test case. It was seen that the pattern images had a low compression ratio of an average of 1.2543, while the landscape images had a slightly higher compression ratio of an average of 1.6753. The high contrast images displayed the highest levels of compression, with an average ratio of 2.1102.

5.3 *Test Case 3: PSNR and MSE Values of the Different Image Types*

This test case is the most vital test case as it determines how well the implemented system including all three compression schemes, handles the introduction of noise. All three schemes are tested with an image of approximate resolution of 1200×800 pixels, with an error probability increasing from 0 to 100. Both error cases are tested, and one image is chosen from each image type for this test with the Andrej Lisakov image being chosen for the pattern image, the Pietro De Grandi for the landscape and the Ricardo for the high contrast. Appendix D3 presents the tables and graphs for this test case. The results show that for the high contrast and landscape images, the created *Holes*-only algorithm outperforms the *DCT*-only and combined *Holes+DCT* scheme by maintaining a higher PSNR. The pattern image showed that the *DCT*-only scheme proved the better option until the error probability was greater than 20%. At that point, the *Holes*-only scheme yielded better results. The MSE for both bit flip and ones compliment error introduction yielded results as predicted for all schemes. This means that with a higher chance of an error occurring, the MSE will increase in proportion, with the ones compliment error introduction increasing at a higher rate.

5.4 *Test Case 4: Testing the Designed Schemes with known Images*

The final test case compares the compression ratio, PSNR and MSE of known images such as mandrill, peppers and cameraman, using the *Holes*-only, *DCT*-only and a Discrete Wavelet Transform (DWT) compression scheme. The main focus was to see how well the *Holes*-only scheme performs. The error probability used was 0% and 10% with the bit flip error introduction chosen. It can be seen that the DWT scheme outperforms the prior schemes with regards to a compression ratio. However, the created holes algorithm yields significantly better results in terms of PSNR with no error being introduced. When errors are introduced, the DCT scheme yields the best PSNR results, however the DWT scheme yields better results in compressing the image, with the trade off of sacrificing image quality.

6. CRITICAL ANALYSIS AND RECOMMENDATIONS

The created compression scheme of creating holes, encoding and filling the holes, is able to compress a variety of images with a positive compression ratio. It can be seen to outperform the compression scheme of DCT and DWT when the probability of an error being introduced increases. The overall system was designed to be functional and achieve all objectives set out for it. However, this was a major trade-off, as the system is implemented so that it is guaranteed to work, and as a result of this no parallel programming was implemented. With the entire implementation being completed serially, the second trade-off for functionality was processing time. A full high definition picture was processed in approximately 1 hour and 10 minutes. This is a weakness in the implementation as current methodologies do not process and compress images in such long periods of time. The *hard-coded* value of 6 when the Chebyshev distance is calculated for creating and filling holes is flawed as it limits the compression depth of the DCT implementation to a value of 7. Since the designed system was based on various existing schemes, colour images are forced to be converted into grayscale as the scheme is unable to handle colour images. Overall, it seems that the created compression scheme is able to give a positive compression ratio while maintaining high image quality.

For future development the designed system can be improved drastically by implementing parallel programming, as this can reduce the processing time of the system significantly. This will allow higher resolution images to be tested against the system and compared to current schemes. Furthermore, a neural network can be trained on multiple images in order to build a large database. This neural network will be able to dictate the value used for Chebyshev distance calculation, that can allow the maximum number of holes to be created and filled in the image. Subsequently, this will also allow the neural network to determine the correct values should the compression depth of the combine scheme with DCT be chosen to be less than 7 [10]. The trained neural network could potentially reconstruct an image of low resolution to an image of higher resolution by maintaining detail due to the way the holes are filled [10]. A final recommendation would see the use of error checking that could resolve any errors that were introduced to the image through the channel.

7. CONCLUSION

The research, design and implementation of an image compression scheme that involved: creating holes in an image, encoding and transmitting the image with random errors being introduced and filling in the respective holes was presented. The overall scheme is able to compress a variety of images and is able to give a fair level of compression while maintaining the quality of the image. The scheme was implemented in the *MATLAB* environment, and is able to compress images of various types. The *Holes* only scheme was able to out-perform the *DCT* only and combined schemes when the probability of an error being introduced increases. Various pattern, landscape and high contrast images were tested against the implemented schemes in a variety of test cases that tested the compression ratio, PSNR, MSE and was compared to existing schemes such as DWT compression. The system is critically analysed with future recommendations given involving the idea of parallel programming and neural networks.

ACKNOWLEDGEMENT

The author would like to thank Professor F. Takawira for his guidance, support and advice throughout the project and his colleague, Mr Nitesh Nana.

REFERENCES

[1] Wei; *An Introduction to Image Compression*; Graduate Institute of Communication Engineering; National Taiwan University; Taipei, Taiwan, ROC

[2] Hussain et al.; *Image Compression Techniques: A Survey in Lossless and Lossy Algorithms*; Journal of Neurocomputing; Vol. 300; March, 2018

[3] Singh et al.; *Various Image Compression Techniques: Lossy and Lossless*; International Journal of Computer Applications; Vol. 142; Pages 23-26; May, 2016

[4] Yoon, Chung ; *EE368B Texture Image Extrapolation for Compression*; `https://web.stanford.edu/class/ee368b/Projects/yoons1/index.html`; Last Accessed: 02/09/2019

[5] Cabeen, Gent; *Image Compression and the Discrete Cosine Transform*; Math 45; College of the Redwoods

[6] Bansal, Gupta, Tomar; *The Implementation of Run Length Encoding for RGB Image Compression*; International Journal of Advanced Research in Computer Engineering & Technology (IJARCET); Vol. 3; Issue 12; December, 2014

[7] Selim et al.; *A Simplified Fractal Image Compression Algorithm*; Journal of the 2008 International Conference on Computer Engineering and Systems, ICCES 2008; Pages 53-58; December, 2008

[8] IEEE Morocco Section; *Distances in Classification*; Café Scientifique; 07 January 2016

[9] Kolodzy; *Cognitive Radio Technology (Second Edition*; Chapter 2 - Communications Policy and Spectrum Management; Pages 27-64; 2009

[10] Hui; *GAN What is Generative Adversary Networks GAN? - Jonathan Hui - Medium*; `https://medium.com/@jonathan_hui/gan-whats-generative-adversarial-networks-and-its-appl` Last Accessed: 05/09/2019

Appendices

A REFLECTION

The Appendix presented here contains a reflection on the project, giving both a personal reflection and how the work was divided among the engineers in the group. Section A1 gives the personal reflection of the project by the author, while Section A2 gives a reflection on the group work done and a table showing how the work was divided.

A1 Personal Reflection

The project required engineers to work in groups of two, and this proved a valuable, enjoyable and highly worthwhile experience. The work was done equally with multiple ideas being presented and combined together so that the overall output of the project would be a success. Kishan and Nitesh had both agreed that since the project was implemented fully in software it would make sense to utilize GitHub to be able to work on the project at the same time. Kishan and Nitesh worked on their own respective computers allowing the project to be implemented quickly, efficiently and be constantly ahead of schedule. This was something that both engineers agreed on was that an attempt must be made to constantly work ahead of schedule in order to compensate for any possible errors occurring.

The initial stages of the project implementation posed no problems to Kishan or Nitesh with ideas and algorithms being developed and finely tuned to produce results of the best calibre. The Gantt chart created in the Project Planing phase of the project was adhered to daily to ensure that certain tasks were being completed ahead of schedule or on time. When problems began to arise such as a lack of understanding in specific areas of research, Nitesh was available on hand to assist Kishan with the understanding so that the work done would be together and to ensure both engineers understand every step of the project. Professor F. Takawira was available for consultation with the engineers and was able to assist in helping the engineers' thought process and reasoning for the project.

Kishan and Nitesh behaved professionally and ethically through the duration of the project producing work of the highest calibre. All meetings were on time and both engineers kept to the allocated time slot and conducted themselves with the utmost respect for each other and Professor F. Takawira.

In conclusion, working in a group provided Kishan with enough experience and understanding on how to interact better with a colleague being able to work with one another, and being able to common agreements or compromises. The work was completed together with two areas being done individually due to the better understanding of that engineer for that topic.

A2 Group Work

Table 1: Table showing how the work was divided between the engineers

Task	Engineer 1 (Kishan Narotam (717 931))	Engineer 2 (Nitesh Nana (720 064))
Research	x	x
Design	x	x
Reading the image	x	x
Converting to grayscale	x	x
Creating the domain pool	x	x
Creating holes in the image	x	x
DCT	x	x
Run Length Encoding/Decoding	x	
Filling the Holes	x	x
Reconstructing the image	x	x
DWT		x
Generating results	x	x
Documentation	x	x

B PROJECT SPECIFICATION OUTLINE

This Appendix presents the original, unmarked Project Specification Outline which can be found on the following two pages.The Project Specification Outline is one the preliminary documents that were required in the Project Planning phase of the overall project. The document gave an outline of the project, a detailed specification for the project. Milestones, budget and resources are discussed while exploring possible risks or mitigations that may occur.

School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
ELEN4002/4012: Project Specification Outline

To be completed by supervisor
Assessment:
☐ Deficient ☐ Acceptable
☐ Good ☐ Excellent

Project Title: Image compression based on non-parametric sampling in noisy environments

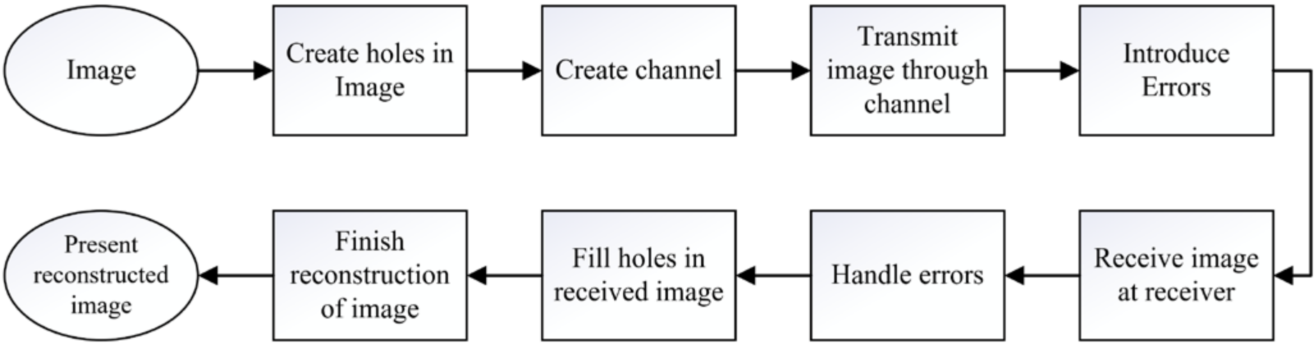
Group Number:	19G01	Supervisor Name:	Prof. Fambirai Takawira
Student Name A:	Kishan Narotam	Student Name B:	Nitesh Nana
Student Number A:	717 931	Student number B:	720 064

Ethics:
☐ Request for waiver (does not involve human participants or sensitive data)
☐ Copy of ethics application attached (Non-medical) – School Committee
Supervisor Signature ☐ Copy of ethics application attached (Medical) – University Committee

Project Outline: *(give a brief outline such that ethics reviewers understand what will be done, 100 words maximum)*

A variety of images will be transmitted via a channel and received after being processed in noisy environments, with the ultimate goal of image compression. The proposed method includes creating holes within the image and transmitting through the channel. The image is received at the receiver and processed in order to fill the holes resulting in the structure of the original image.

Project Specification:



An image will be chosen for compression utilizing the compression technique of creating holes in the image and after transmitting, filling of those holes will result in the original image.
(The creation of holes results in the compression of the original image).
The holes will be created manually making use of current compression algorithms (such as fractal compression for example). Once the holes are created, the image will be transmitted via a simple channel that will introduce errors randomly.
The image will be received at the receiver, and knowing where the holes are present, will fill them accordingly using a different algorithm. The receiver will also have to deal with random errors that may have occurred from the channel. The received image with the filled holes will be reconstructed and presented as the final image which should coincide with the original image.

Should time be permitting, alternate compression algorithms (e.g. DCT) will be utilized on the same image and used as a way of comparing the created holes compression technique with a common practice compression technique (similar to DCT for image compression for JPEGs).

Milestones:

The initial goal would be to complete a literature review before the commencement of the project and have a plan on which specific algorithms will be implemented.

The algorithm for creating holes in the image will take roughly one and a half weeks to complete. This will be completed in parallel with setting up the channel necessary for transmitting images.

Introducing errors is estimated to take half a week of time.

Handling errors and filling of holes will be completed in parallel and is estimated to take around 2 and a half weeks to complete.

One week is set out for debugging and refining of algorithms,

The remaining time is allocated to documentation and report writing.

Some tasks may be divided between members in order for them to be completed in parallel. For example, each person may work individually on different parts of the holes algorithm and combine workings later on.

Preliminary Budget & Resources:

Budget:
No budget is required and personal laptops/computers will be utilized.

Resources:
Since this project requires no physical hardware building, only computers will be utilized in implementing the algorithms and processing the images.

The various programs used to implement the algorithms will require the relevant software licenses (most probable is MATLAB). Student licenses provided by the university.

Processing the images, will require the actual images (image data), these could range from textured images to some more complex.

Risks / Mitigation:

- Using images available from the web may lead to intellectual property and copyright infringements. To mitigate this risk, only approved data labeled for reuse will be utilized.
- University protest action may cause unforeseen interruptions. Having laptops and due to the nature of the project, working on campus grounds is not an absolute necessity and work may be completed in a different environment.
- Eskom's current power crisis may result in data corruption as everything will be done on computers. However, since laptops will be utilized, this may be prevented, but laptops have a limited battery life and so work must be saved throughout the process of image compression and broken down into smaller, more manageable segments that can be completed in shorter cycles.

C PROJECT PLAN REPORT

This Appendix present the original, unmarked Project Plan Report which can be found on the following 7 pages. The Project Plan Report is the second preliminary document that is required as part of the Project Planning phase for the overall project. The report details the background of the project giving a brief literature review. The proposed strategy for the project is discussed in detail with algorithms being presented along with formulas and supporting images to show the idea. A performance index is discussed with cost and time management in conjunction with how documentation will be done.

PROJECT PLAN
Image Compression based on Non-Parametric Sampling in Noisy Environments

Group 19G01: Kishan Narotam (717 931) & Nitesh Nana (720 064)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: This report covers the project plan for the laboratory project based on image compression. The project plan looks at the specifications that will define a successful project implementation and a brief look at existing models of image compression. The proposed strategy involves an encoder and decoder side that will be implemented in MATLAB. The encoder side of the strategy involves: reading the image, converting to greyscale, dividing the image into a domain pool, creating holes in the image and compressing the image using known technique. The image is encoded and transmitted through a channel and random errors introduced. The decoder involves: decompressing the image, filling the holes and reconstructing the final image. A performance index is proposed as well as a look into cost management. Time management is discussed in detail with the aid of a Gantt chart and how the documentation of the project will be done.

Key words: Image compression, DCT, MATLAB, Project Plan, Hole creation

1. INTRODUCTION

A project plan is a formal document that forms a guideline of the project that will ensure the overall success and completion of the project [1]. Planning the project involves looking at three fundamental aspects of the project:

- Scope: what are the objectives of the project and how it will be completed
- Cost: how much money is allocated, budgeted and spent over the course of the project
- Time: the time taken to execute the project to a point of completion [2].

In the report that follows, a project plan is created for the telecommunications project of *image compression based on non-parametric sampling in noisy environments*. Firstly, the project specifications are defined, followed by a brief look into existing models. Subsequently, the proposed strategy for the project is done with a cost and time management which is discussed in order for the project to be completed on time within a specified budget.

2. PROJECT SPECIFICATIONS

The aim of the project is to create a robust scheme for determining multiple holes that may be received in an image and appropriately fill these holes. In addition, the image may be subject to random burst errors which need to be identified and corrected.

An image will be chosen for compression utilizing a compression technique and creating holes in the image. After transmission, filling of those holes will result in the original image. The holes will be created manually with a proposed algorithm and the additional compression layer will make use of current compression algorithms (DCT or fractal compression). Once the holes are created, the image will be transmitted via a simple channel that will introduce errors randomly. The image will be received at the receiver, and knowing where the holes are present through the encoding scheme will fill them accordingly using a different algorithm. The receiver will also have to deal with random errors that may have occurred from the channel. The received image with the filled holes will be reconstructed and presented as the final image which should coincide with the original image.

3. EXISTING MODELS

Image compression is an application of data compression where an image file is encoded with a few bits with the overall goal of reducing the size of the image file compared to that of the original [3]. There are two main techniques of image compression:

- Lossless
- Lossy

3.1 Lossless

Lossless compression allows the original form of data to be reproduced, thus meaning that the original data from the file before compression is preserved [4]. Some of the current lossless compression techniques include:

- Run-Length encoding
- Huffman encoding
- Shannon-Fano encoding
- Arithmetic
- Dictionary based [5].

3.2 Lossy

Lossy compression removes some of the data from the original file, resulting in an overall reduction of the size of the file [4]. The non-useful parts of the data that is not noticeable is removed, thus reducing the overall quality of the data and file. Some of the current lossy compression techniques include:

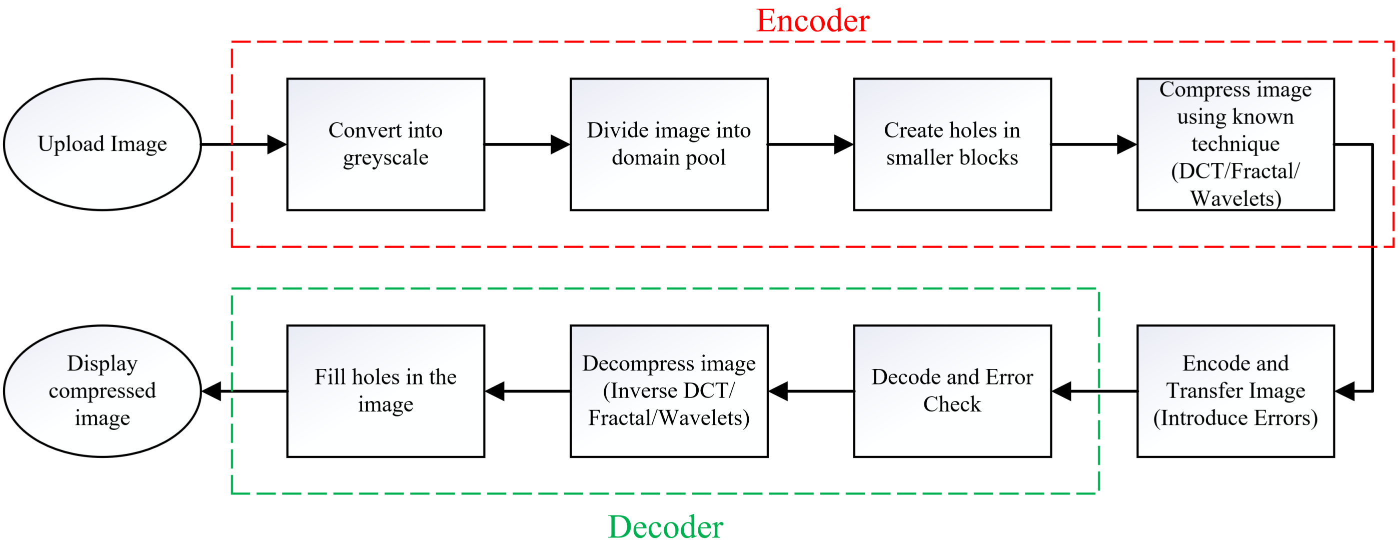


Figure 1 : Block diagram of the proposed strategy

- Lossy predictive
- Vector Quantization
- Transform Coding
- Block transform
- DCT/DWT
- JPEG [5].

4. PROPOSED STRATEGY

The proposed solution will be implemented in MATLAB and can be broken down into two main facets:

- Encoder side:
  - Loading/Reading an Image into MATLAB
  - Converting the image to a greyscale image
  - Dividing the image into the domain pool
  - Creating holes in the smaller blocks
  - Compressing the image using a known technique (DCT/wavelets/fractal)
- Decoder side:
  - Decode the image and error check
  - Decompress image (inverse DCT/wavelets/fractal)
  - Fill holes in the image

In conjunction with this, a simple channel will be created and random errors will be introduced after the image is encoded. Figure 1 shows the basic framework and block diagram of the proposed solution that will be implemented.

5. ENCODER SIDE

5.1 *Step 1: Loading/Reading an Image into MATLAB*

Firstly, an image is loaded into MATLAB, and a PNG or BMP image is specifically chosen with the dimensions $M \times N$, where M and N are divisible by 8. A PNG or BMP image is used as a JPEG image, is already a compressed image via the DCT image compression technique. Figure 2 shows an example of the image that can be used.

Figure 2 : 128×128 image that will be loaded

5.2 *Step 2: Convert the image to greyscale*

The loaded image is converted to greyscale. The reason for this is because an image that has been imported into MATLAB creates a 3-dimensional array where the first two elements of the array represent the dimensions of the image and the third dimension is the colour map. Converting the image to a greyscale one, creates a 2-dimensional array, which are simply the dimensions of the image, where each array value correlates to the specific integer value of that specific pixel. Figure 3 shows an example of the outcome of converting the image to greyscale.

Figure 3 : Image after it has been greyscaled in MATLAB

5.3 Step 3: Divide the image into the domain pool

The image is now divided into smaller 8×8 blocks, creating the domain pool. Once the image has been divided into smaller blocks, we index each block from the top left starting from 1 and increment each index by one, moving left-to-right, top-to-bottom. The reason for creating such an index is so that the receiver will be able to determine which smaller block in the domain pool contains holes so that the receiver can fill these holes. Figure 4 shows an example of the loaded image having a domain pool of 256 blocks.

5.3.1 Indexing the blocks Since the images that will be chosen will have dimensions divisible by 8, the first step in determining how many blocks in the domain pool will be created, would be to divide the width and height of the image by 8 to determine how many blocks there will be going across and down the image. Algorithm 1 shows a very high-level approach to finding the top left pixel of a block in the domain pool

5.4 Step 4: Create holes in the smaller blocks

With the domain pool created, holes are created by starting at the center square within the 8×8 block. Starting off with the center 2×2 square, the average value of those 4 pixels are calculated. Each pixel in the 2×2 square is compared to the average value of the smaller square using the Chebyshev distance. The Chebyshev distance between each pixel and the average forms the basis of the similarity index that will be used in the project and its implementation. If the value of the similarity index is less than 5 for all pixels, a hole can be created in those 4 pixels. A larger square in the same 8×8 block is checked with a size of 4×4. The average of these pixels are calculated and as before compared to each value of the center square. This can continue until we reach a center square size of 6×6, which we will define as the largest hole that can be inserted. Algorithm 2 shows a high-level algorithm for creating the holes in the blocks in the domain pool.

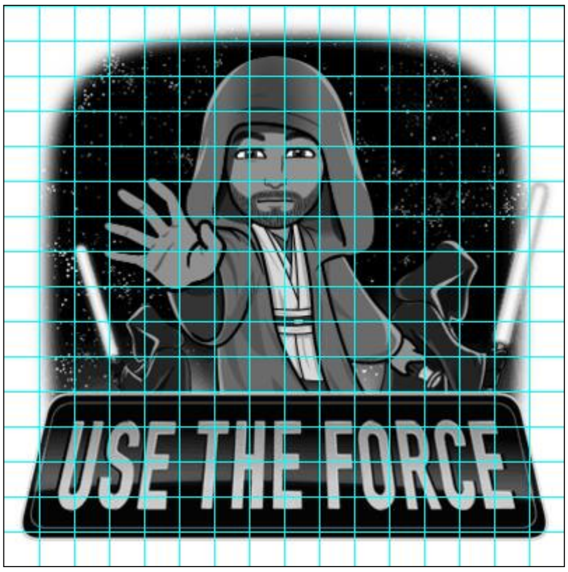


Figure 4 : Smaller 8×8 blocks creating the domain pool

Algorithm 1 High level algorithm of finding the `x,y` coordinates of the respective blocks in the domain pool

```
xBlocks ← widthOfImage/8
yBlocks ← heightOfImage/8
Set initial (x, y) values to be (1, 1) for block 1
To move onto next block:
  set x to 8(i) − 7, where i is the block number
if BlockNumber > xBlocks then
   Set y to 8(i + 1) − 7
   Reset x to be 1 and move to next block
end if
```

Figure 5 shows how each of the 8×8 squares in the domain pool, will be checked, starting with the blue squares, moving up to the orange squares to the largest defined hole which are the green squares.

5.5 Step 5: Compress image using a known technique (DCT/Fractal/Wavelets)

The DCT compression technique will be utilized for the compression of the image. DCT will be performed on each block in the domain pool. DCT is implemented on a 8×8 matrix, which is the reason the domain pool is made up of multiple 8×8 blocks and that an image where the dimensions are divisible by 8 is chosen. The image is now compressed and can be

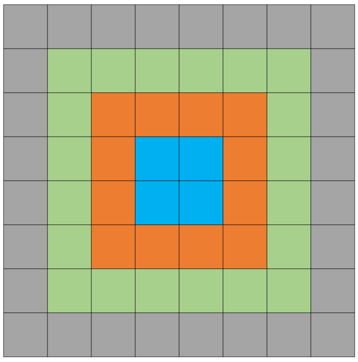


Figure 5 : How a 8 × 8 block (grey square) will be checked for hole creation starting with the blue square.

transmitted via the channel. MATLAB has a function built-in to perform the transform, however Equation 1 shows the mathematical representation of the transform.

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (1)$$

5.6 Step 6: Encode and Transfer image

The image is encoded as a binary message, and when transmitted through the channel, errors are randomly introduced. The data that will be sent through the channel will be a bit stream and the errors introduced will be based off of a byte with a probability of 10^{-3} where the byte can then be randomized. Alternatively each bit can be randomly selected, using the same probability above, to be prone to an error and the bit value flipped.

6. DECODER SIDE

6.1 Step 7: Decode and Error Check

Identification of the location of the errors is required. First, identifying the errors are required, and if detected must be corrected and then the image can be decoded. The image will be converted from a bit stream into its 2-dimensional matrix.

6.2 Step 8: Decompress (Inverse DCT/Fractal/Wavelets)

Since DCT compression was used, we now perform the inverse DCT to obtain the compressed image and its respective values in the 2-dimensional matrix.

Algorithm 2 High level algorithm of method of creating holes

```
x coordinate ← x+3
y coordinate ← y+3
Calculate average of squares: (x,y; x+1,y; x, y+1; x+1, y+1)
for x till x+1 do
    for y till y+1 do
        Check Chebyshev distance between (x,y) and average
    end for
end for
if Chebyshev distance between average & each pixel ≤ 5 then
    Set value in pixels (x,y; x+1,y; x, y+1; x+1, y+1) to 0
else
    Compress image
end if

x coordinate ← x+2
y coordinate ← y+2
Calculate average of squares (now 4 × 4 square)
for x till x+3 do
    for y till y+3 do
        Check Chebyshev distance between (x,y) and average
    end for
end for
if Chebyshev distance between average & each pixel ≤ 5 then
    Set value in pixels to 0
else
    Compress image
end if

x coordinate ← x+1
y coordinate ← y+1
Calculate average of squares (now 6 × 6 square)
for x till x+3 do
    for y till y+3 do
        Check Chebyshev distance between (x,y) and average
    end for
end for
if Chebyshev distance between average & each pixel ≤ 5 then
    Set value in pixels to 0
else
    Compress image
end if
Compress image
```

6.3 Step 9: Fill holes in the image

With the one-dimensional array sent across the channel, the blocks within the domain pool that contain the holes can be identified. Starting at outer pixels, the average pixel value of the square is calculated, and that will fill the inner pixels where the hole was created.

7. PERFORMANCE INDEX

The nature of this project is based around current methodologies and is more research-based. Results are required to present the overall performance of the proposed technique. MATLAB allows for relative simplicity in creating and performing many image compression techniques such as DCT. For that reason, other compression techniques such as fractal compression or wavelets compression will be implemented in addition to DCT compression. The DCT compression will be set as the benchmark and a comparison with the fractal or wavelets compression will be completed. The values of signal-to-quantization-noise ratio (SQNR), peak-signal-to-noise-ratio (PSNR) and mean-squared-error (MSE) will be the main focus of comparison.

8. COST MANAGEMENT

The nature of this project is software based, and thus the total costs are minimal to none. Licences for the MATLAB software is already provided in the form of a student licence by the university, and personal computers or laptops will be used.

9. TIME MANAGEMENT

The Project window runs from 01 July 2019 until 11 September 2019. The proposed timeline of events and duration of these events can be seen in Figure 6, in conjunction with this, Figure 7 shows the . The project begins with the planning phase including a literature review and the development of the project plan. The overall idea and method was continuously discussed between the team of engineers and the associated supervisor. Once the project plan is approved, the engineers can begin with the execution of the plan.

The encoder side is tackled first while the channel can be created in parallel with that. Once the encoder and channel have been completed and tested, the decoder side of the proposed solution can be implemented. The initial implementation of the encoder will utilize DCT, and once an entire solution is created and tested using DCT, other compression techniques can be implemented in conjunction with the proposed solution. The results of the DCT compression is compared to the other implemented compression technique.

Based on the planned approximated time that is allocated to each task, this indicates that keeping to a tight schedule is required. Since the supervisor will not be available during the first two weeks, the engineers will have to work together and communicate efficiently in order for them to encounter minimal problems and build the proposed solution.

10. DOCUMENTATION

Throughout the project window, constant meetings and discussions will be held between the engineers and the supervisor. As seen in Figure 6 and 7 the *Monitoring and Control* task is run from the day after the project plan is submitted till the project is presented. All meetings have and will be documented as meeting minutes, discussing the meeting and the agenda of the project at that period of time. A GitHub repository is created so that the engineers can constantly work on the proposed strategy while keeping all documents and helpful materials available to them at all times.

11. CONCLUSION

The proposed project plan was given approval by the supervisor. This document discussed existing models of image compression, and gives a detailed description of the proposed strategy of image compression for this project. Block diagrams, example images and high-level algorithms are given as a way of explaining the proposed strategy in more detail. Multiple image compression techniques will be utilized, and the results will be discussed as a performance index. The cost management is discussed, however since the project is research and software-based, no costs will be incurred in the proposed solution. The time management of the project is discussed in detail with a Gantt chart presented as supporting material.

REFERENCES

[1] Techopedia; *What is a Project Plan? - Definition from Techopedia*; `https://www.techopedia.com/definition/24775/project-plan`; Last Accessed: 02/07/2019

[2] Heerkins, G R; *Project Management*; McGraw-Hill; New York, NY, United States; 1st Edition, 2001

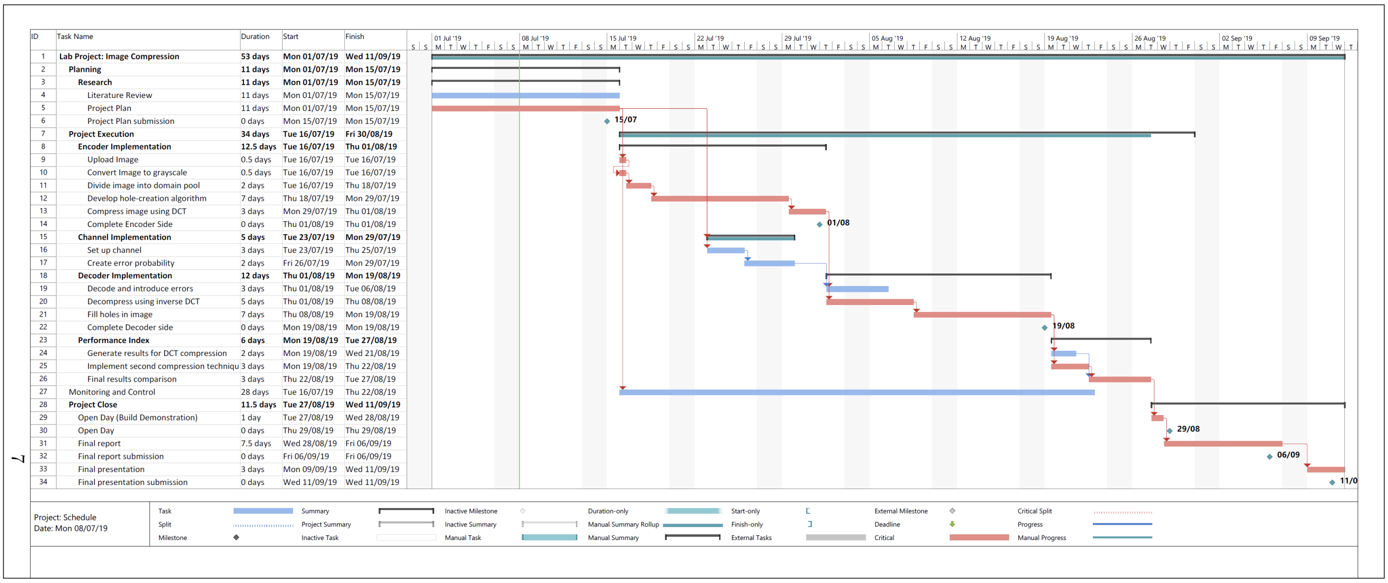
[3] Wei-Yi Wei; *An Introduction to Image Compression*; Graduate Institute of Communication Engineering; National Taiwan University; Taipei, Taiwan, ROC

[4] Chapman, C; *Everything You Need to Know About Image Compression — The JotForm Blog*; `https://www.jotform.com/blog/everything-you-need-to-know-about-image-compression/`; Last Accessed: 02/07/2019

[5] Singh, M; Kumar, S; Chouhan, SS; Shrivastava, M (PhD); *Various Image Compression Techniques: Lossy and Lossless*; Journal of International Journal of Computer Applications; Volume 142; Issue No. 6; pp 23 - 26; May 2016

	Task Name	Duration	Start	Finish	Predecessors
1	**Lab Project: Image Compression**	**53 days**	**Mon 01/07/19**	**Wed 11/09/19**	
2	**Planning**	**11 days**	**Mon 01/07/19**	**Mon 15/07/19**	
3	**Research**	**11 days**	**Mon 01/07/19**	**Mon 15/07/19**	
4	Literature Review	11 days	Mon 01/07/19	Mon 15/07/19	
5	Project Plan	11 days	Mon 01/07/19	Mon 15/07/19	
6	Project Plan submission	0 days	Mon 15/07/19	Mon 15/07/19	
7	**Project Execution**	**34 days**	**Tue 16/07/19**	**Fri 30/08/19**	
8	**Encoder Implementation**	**12.5 days**	**Tue 16/07/19**	**Thu 01/08/19**	
9	Upload Image	0.5 days	Tue 16/07/19	Tue 16/07/19	5
10	Convert Image to grayscale	0.5 days	Tue 16/07/19	Tue 16/07/19	9
11	Divide image into domain pool	2 days	Tue 16/07/19	Thu 18/07/19	10
12	Develop hole-creation algorithm	7 days	Thu 18/07/19	Mon 29/07/19	11
13	Compress image using DCT	3 days	Mon 29/07/19	Thu 01/08/19	12
14	Complete Encoder Side	0 days	Thu 01/08/19	Thu 01/08/19	
15	**Channel Implementation**	**5 days**	**Tue 23/07/19**	**Mon 29/07/19**	**5**
16	Set up channel	3 days	Tue 23/07/19	Thu 25/07/19	5
17	Create error probability	2 days	Fri 26/07/19	Mon 29/07/19	16
18	**Decoder Implementation**	**12 days**	**Thu 01/08/19**	**Mon 19/08/19**	
19	Decode and introduce errors	3 days	Thu 01/08/19	Tue 06/08/19	13,17
20	Decompress using inverse DCT	5 days	Thu 01/08/19	Thu 08/08/19	13
21	Fill holes in image	7 days	Thu 08/08/19	Mon 19/08/19	20
22	Complete Decoder side	0 days	Mon 19/08/19	Mon 19/08/19	
23	**Performance Index**	**6 days**	**Mon 19/08/19**	**Tue 27/08/19**	
24	Generate results for DCT compression	2 days	Mon 19/08/19	Wed 21/08/19	21
25	Implement second compression technique	3 days	Mon 19/08/19	Thu 22/08/19	21
26	Final results comparison	3 days	Thu 22/08/19	Tue 27/08/19	24,25
27	Monitoring and Control	28 days	Tue 16/07/19	Thu 22/08/19	5
28	**Project Close**	**11.5 days**	**Tue 27/08/19**	**Wed 11/09/19**	
29	Open Day (Build Demonstration)	1 day	Tue 27/08/19	Wed 28/08/19	26
30	Open Day	0 days	Thu 29/08/19	Thu 29/08/19	
31	Final report	7.5 days	Wed 28/08/19	Fri 06/09/19	29
32	Final report submission	0 days	Fri 06/09/19	Fri 06/09/19	
33	Final presentation	3 days	Mon 09/09/19	Wed 11/09/19	31
34	Final presentation submission	0 days	Wed 11/09/19	Wed 11/09/19	

Figure 6 : Table showing the tasks and time allocated in completing the project



ID	Task Name	Duration	Start	Finish
1	**Lab Project: Image Compression**	**53 days**	**Mon 01/07/19**	**Wed 11/09/19**
2	**Planning**	**11 days**	**Mon 01/07/19**	**Mon 15/07/19**
3	**Research**	**11 days**	**Mon 01/07/19**	**Mon 15/07/19**
4	Literature Review	11 days	Mon 01/07/19	Mon 15/07/19
5	Project Plan	11 days	Mon 01/07/19	Mon 15/07/19
6	Project Plan submission	0 days	Mon 15/07/19	Mon 15/07/19
7	**Project Execution**	**34 days**	**Tue 16/07/19**	**Fri 30/08/19**
8	**Encoder Implementation**	**12.5 days**	**Tue 16/07/19**	**Thu 01/08/19**
9	Upload Image	0.5 days	Tue 16/07/19	Tue 16/07/19
10	Convert Image to grayscale	0.5 days	Tue 16/07/19	Tue 16/07/19
11	Divide image into domain pool	2 days	Tue 16/07/19	Thu 18/07/19
12	Develop hole-creation algorithm	7 days	Thu 18/07/19	Mon 29/07/19
13	Compress image using DCT	3 days	Mon 29/07/19	Thu 01/08/19
14	Complete Encoder Side	0 days	Thu 01/08/19	Thu 01/08/19
15	**Channel Implementation**	**5 days**	**Tue 23/07/19**	**Mon 29/07/19**
16	Set up channel	3 days	Tue 23/07/19	Thu 25/07/19
17	Create error probability	2 days	Fri 26/07/19	Mon 29/07/19
18	**Decoder Implementation**	**12 days**	**Thu 01/08/19**	**Mon 19/08/19**
19	Decode and introduce errors	3 days	Thu 01/08/19	Tue 06/08/19
20	Decompress using inverse DCT	5 days	Thu 01/08/19	Thu 08/08/19
21	Fill holes in image	7 days	Thu 08/08/19	Mon 19/08/19
22	Complete Decoder side	0 days	Mon 19/08/19	Mon 19/08/19
23	**Performance Index**	**6 days**	**Mon 19/08/19**	**Tue 27/08/19**
24	Generate results for DCT compression	2 days	Mon 19/08/19	Wed 21/08/19
25	Implement second compression techniqu	3 days	Mon 19/08/19	Thu 22/08/19
26	Final results comparison	3 days	Thu 22/08/19	Tue 27/08/19
27	Monitoring and Control	28 days	Tue 16/07/19	Thu 22/08/19
28	**Project Close**	**11.5 days**	**Tue 27/08/19**	**Wed 11/09/19**
29	Open Day (Build Demonstration)	1 day	Tue 27/08/19	Wed 28/08/19
30	Open Day	0 days	Thu 29/08/19	Thu 29/08/19
31	Final report	7.5 days	Wed 28/08/19	Fri 06/09/19
32	Final report submission	0 days	Fri 06/09/19	Fri 06/09/19
33	Final presentation	3 days	Mon 09/09/19	Wed 11/09/19
34	Final presentation submission	0 days	Wed 11/09/19	Wed 11/09/19

Figure 7 : Gantt chart showing the tasks and time allocated in completing the project with the critical path highlighted

D RESULTS

The following sections within this Appendix present all the tables, graphs, and precessed images for the various test cases discussed in Section 5.. Section D1 presents all results for the first test case, where the same image is compressed at different resolutions. Section D2 presents all results relating to the second test case, where the average compression ratio of the different image types are evaluated. The third test case results, in Section D3, show the results for the image types at increasing error probabilities measuring the MSE and PSNR. The final section, Section D4 presents the results for the comparison of the created scheme, DCT and DWT.

D1 Test Case 1: Same Image at Different Sizes

D1.1 Pattern Images

Table 2: Table showing the compression ratio, PSNR and MSE of the different pattern images at different resolutions

Andrej Lisakov				
% Smaller	**Image Size (% of the original image)**	**Compression Ratio**	**PSNR**	**MSE**
0	100	1.3336	34.473003	23.215610
20	80	1.3569	34.512885	23.003390
40	60	1.3886	34.461957	23.274730
60	40	1.5082	34.169225	24.897620
80	20	2.2679	32.929120	33.125950
Bryan Garces				
0	100	1.2468	34.027297	25.724720
20	80	1.2142	34.089827	25.356990
40	60	1.1926	34.112199	25.226700
60	40	1.1947	34.113675	25.218130
80	20	1.3180	33.944174	26.221830
Ross Elder				
0	100	1.4106	34.100531	25.294570
20	80	1.4068	33.858347	26.745190
40	60	1.4194	33.450771	29.376720
60	40	1.5082	33.766507	27.316790
80	20	1.0895	33.005498	32.548470
Vino Li				
0	100	1.0240	34.259220	24.387000
20	80	1.0153	34.273794	24.305300
40	60	1.0090	34.441365	23.385350
60	40	1.0192	34.320747	24.043940
80	20	1.1518	34.079446	25.417670

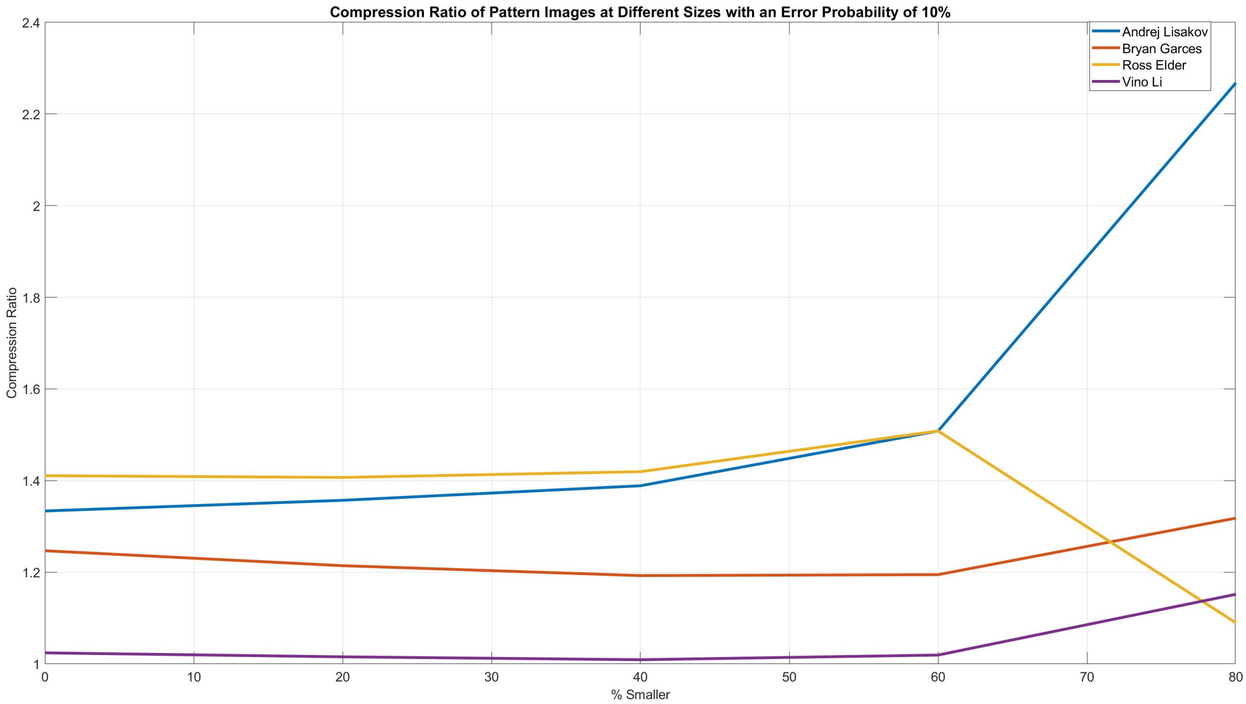


Figure 2: Graph showing the compression ratio against the size of the image for pattern images

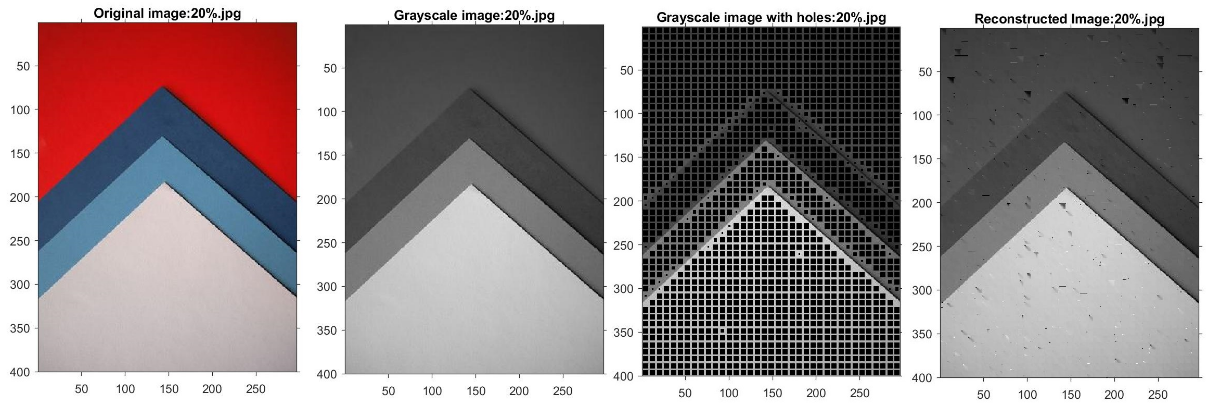


Figure 3: Andrej Lisakov texture image at 20% of the original size with 10% error introduction

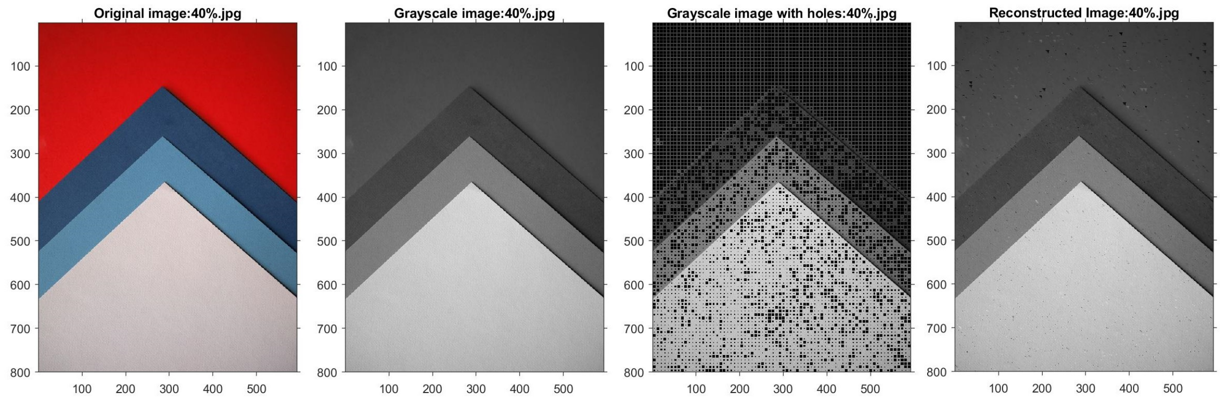


Figure 4: Andrej Lisakov texture image at 40% of the original size with 10% error introduction

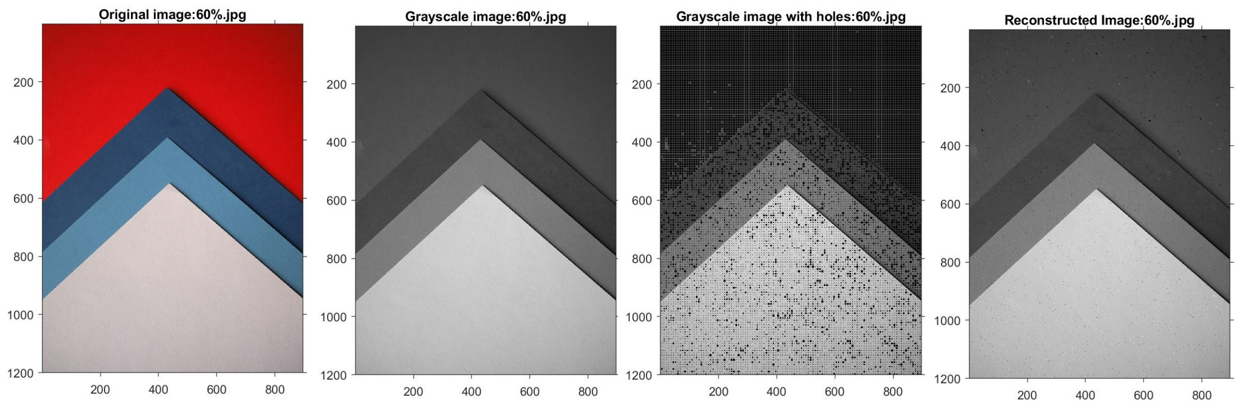


Figure 5: Andrej Lisakov texture image at 60% of the original size with 10% error introduction

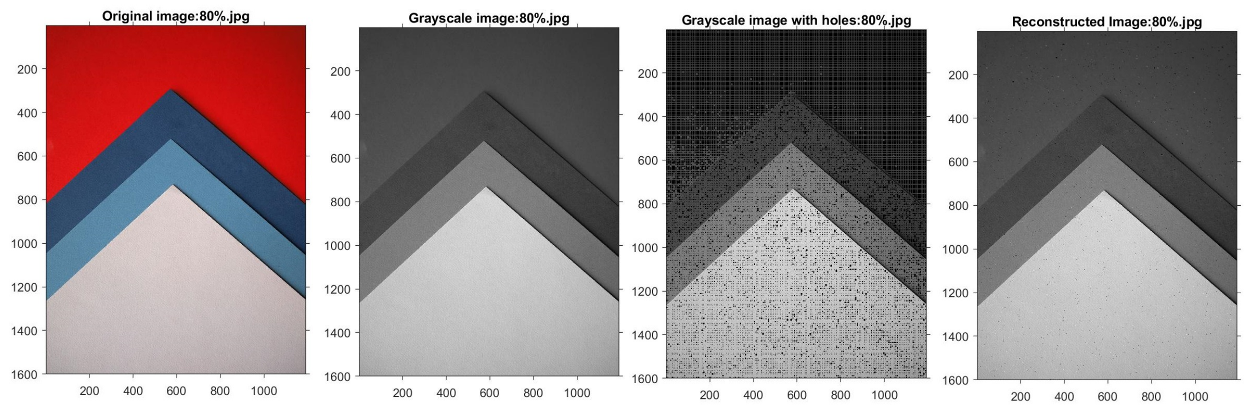


Figure 6: Andrej Lisakov texture image at 80% of the original size with 10% error introduction

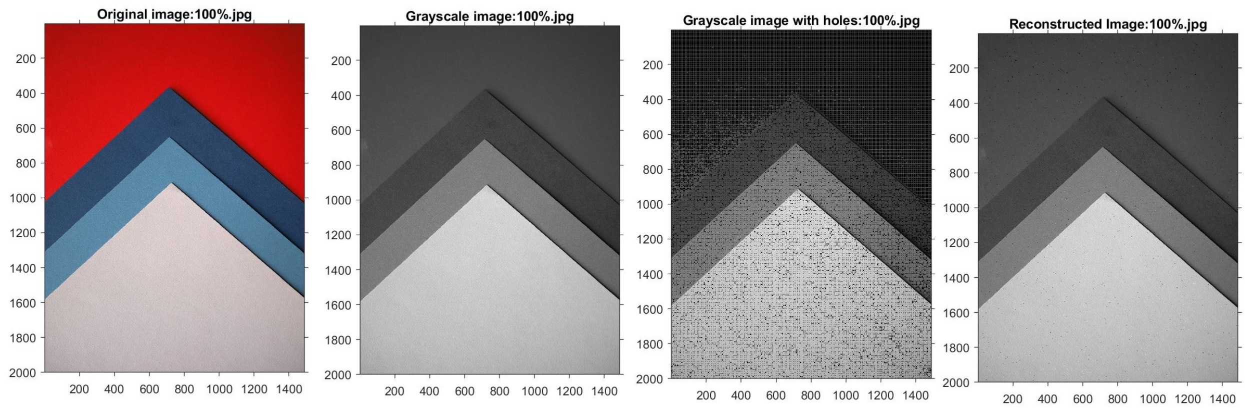


Figure 7: Andrej Lisakov texture image at 100% of the original size with 10% error introduction

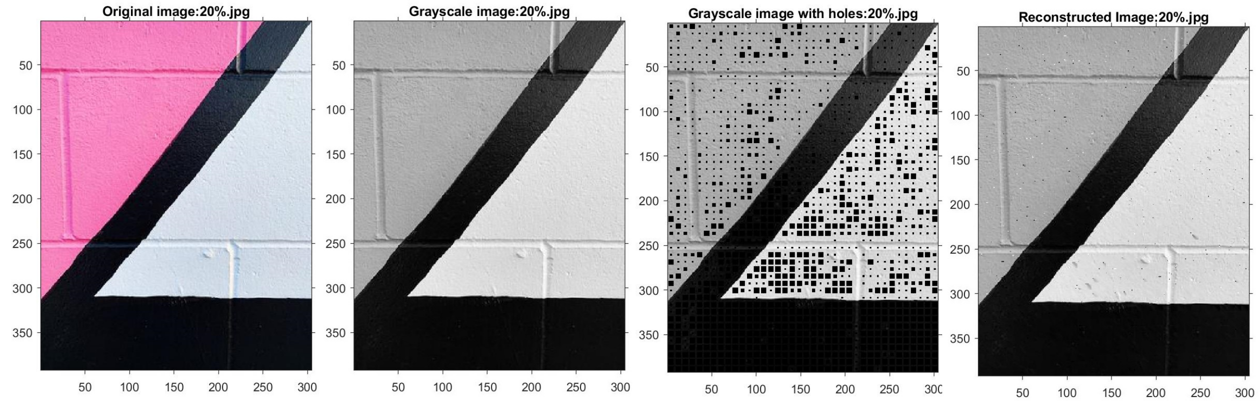


Figure 8: Bryan Garces texture image at 20% of the original size with 10% error introduction

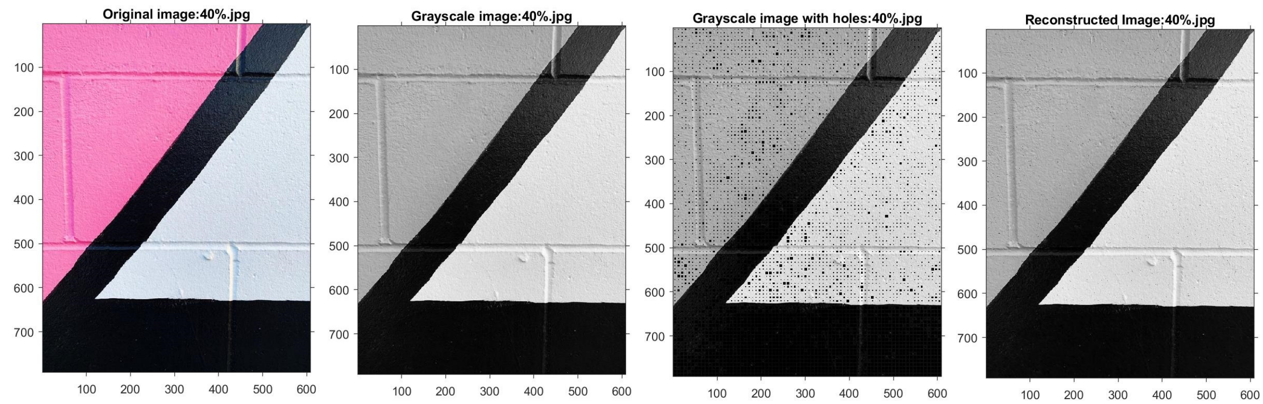


Figure 9: Bryan Garces texture image at 40% of the original size with 10% error introduction

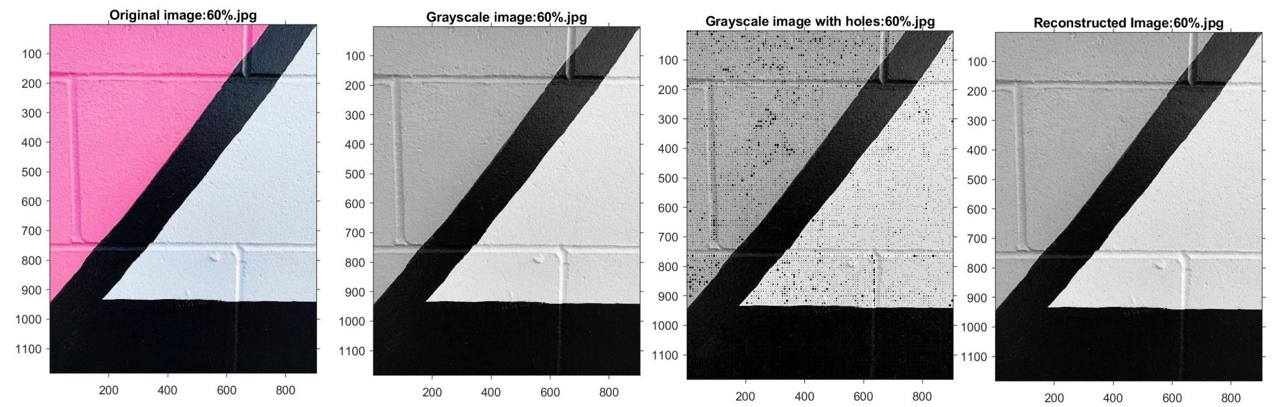


Figure 10: Bryan Garces texture image at 60% of the original size with 10% error introduction

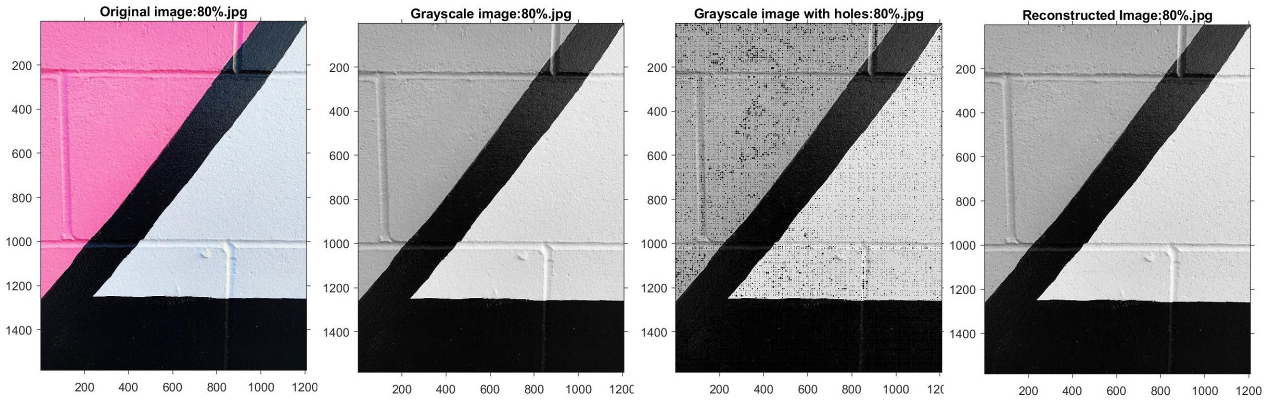


Figure 11: Bryan Garces texture image at 80% of the original size with 10% error introduction

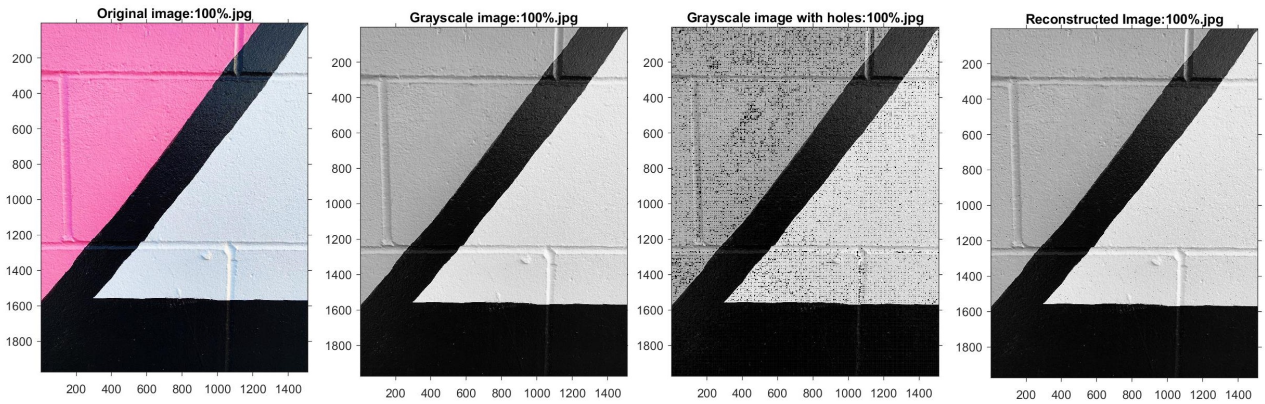


Figure 12: Bryan Garces texture image at 100% of the original size with 10% error introduction

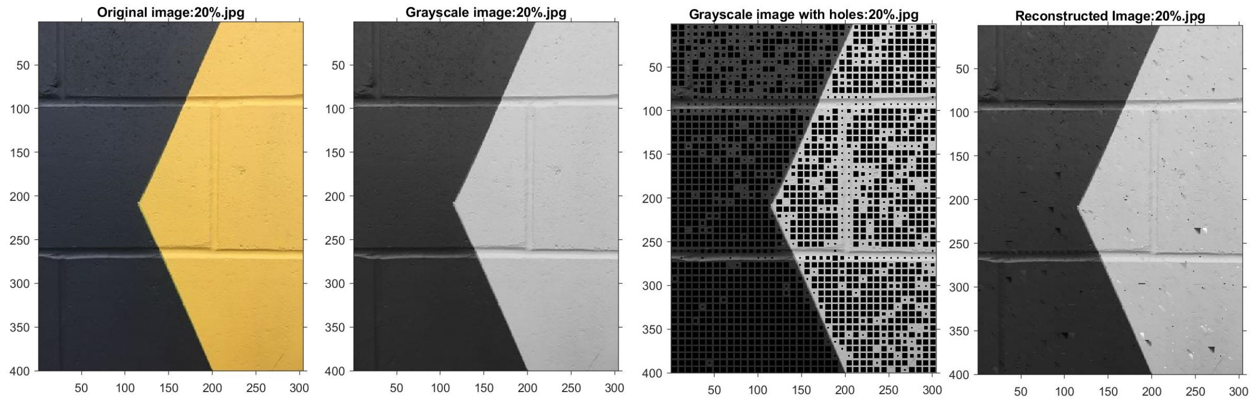


Figure 13: Ross Elder texture image at 20% of the original size with 10% error introduction

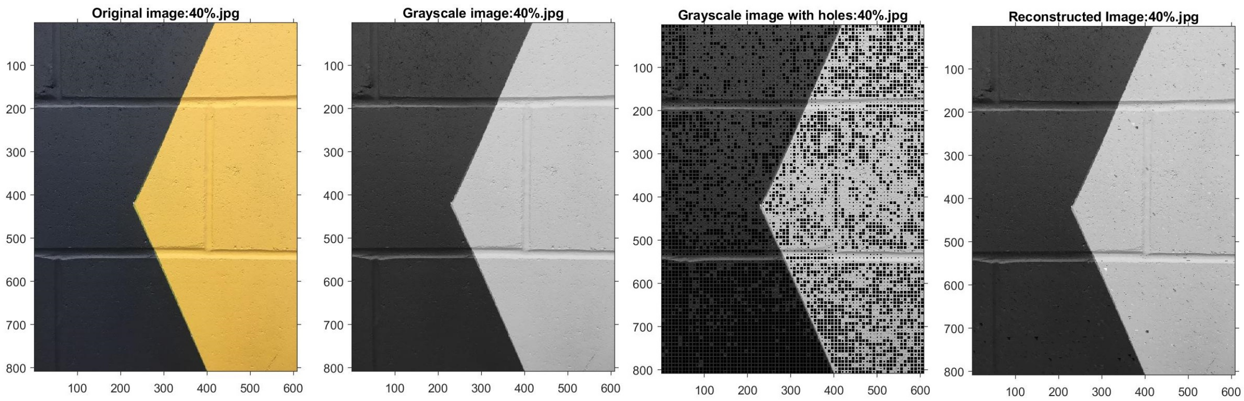


Figure 14: Ross Elder texture image at 40% of the original size with 10% error introduction

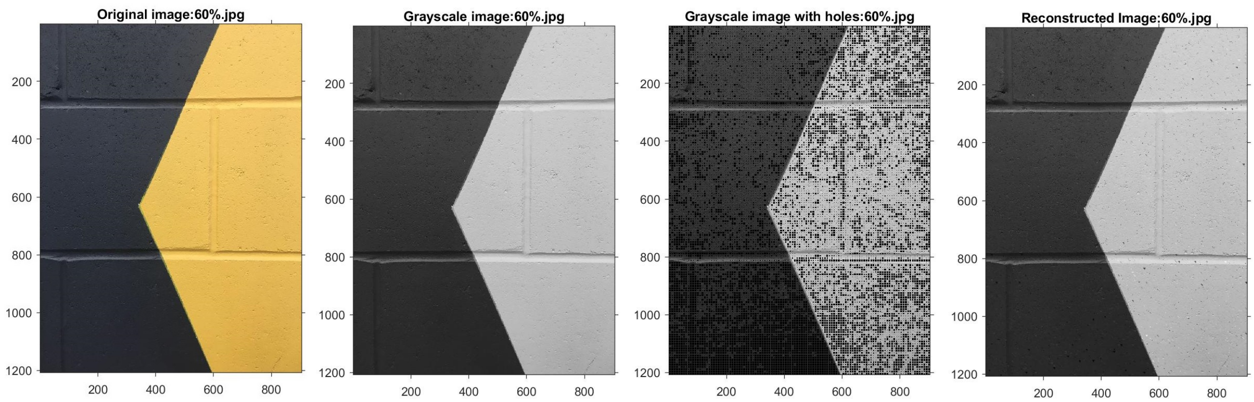


Figure 15: Ross Elder texture image at 60% of the original size with 10% error introduction

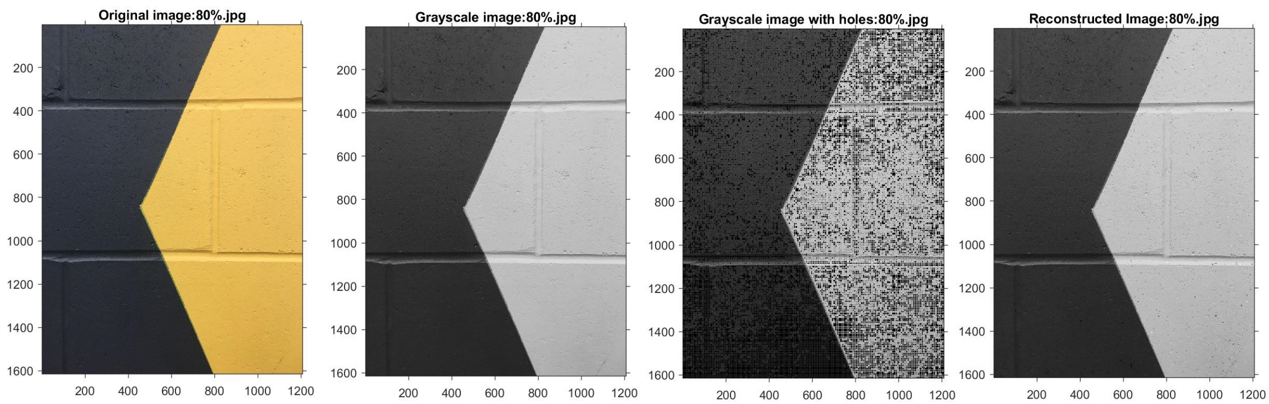


Figure 16: Ross Elder texture image at 80% of the original size with 10% error introduction

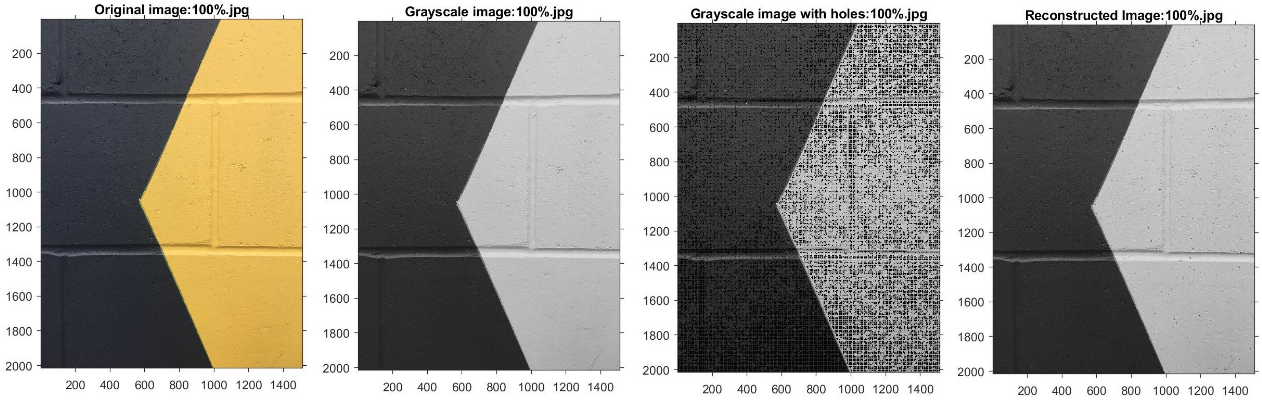


Figure 17: Ross Elder texture image at 100% of the original size with 10% error introduction

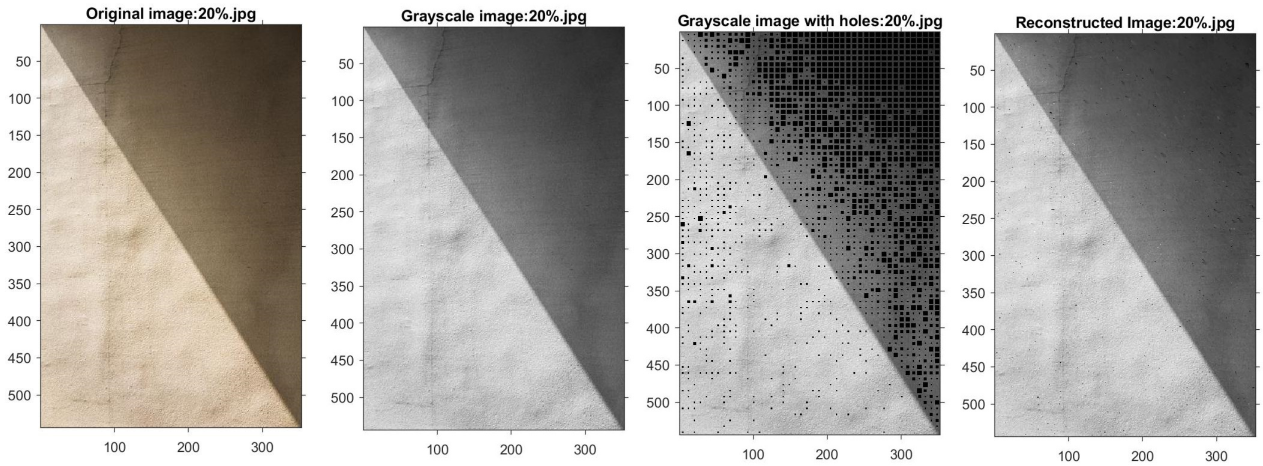


Figure 18: Vino Li texture image at 20% of the original size with 10% error introduction

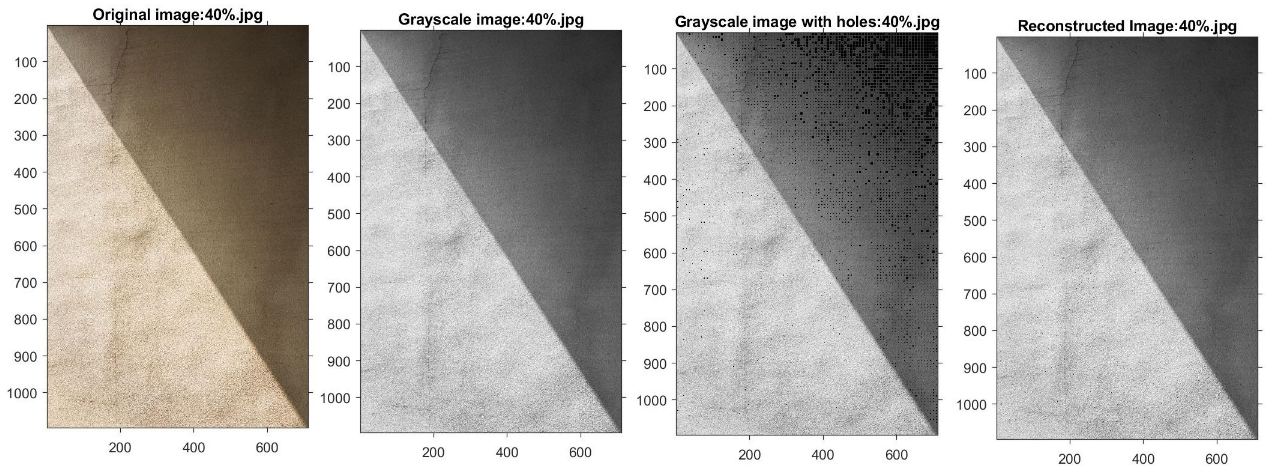


Figure 19: Vino Li texture image at 40% of the original size with 10% error introduction

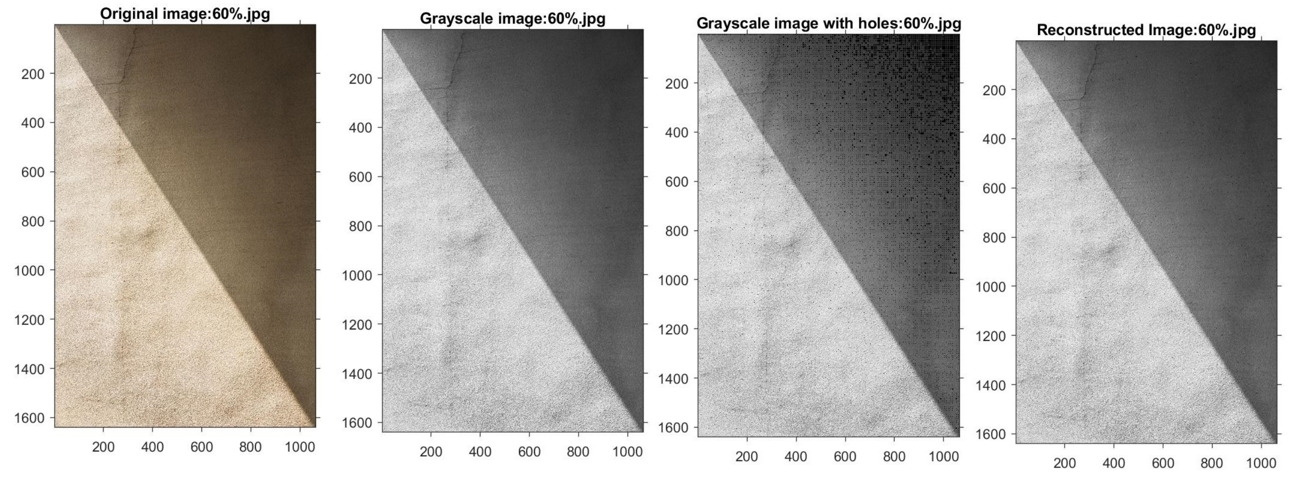


Figure 20: Vino Li texture image at 60% of the original size with 10% error introduction

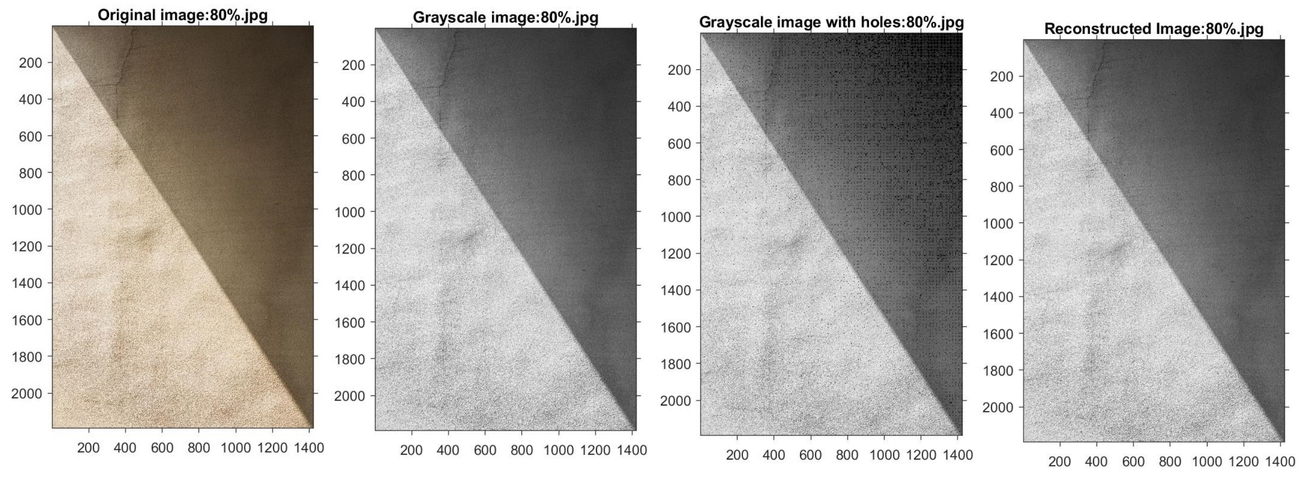


Figure 21: Vino Li texture image at 80% of the original size with 10% error introduction

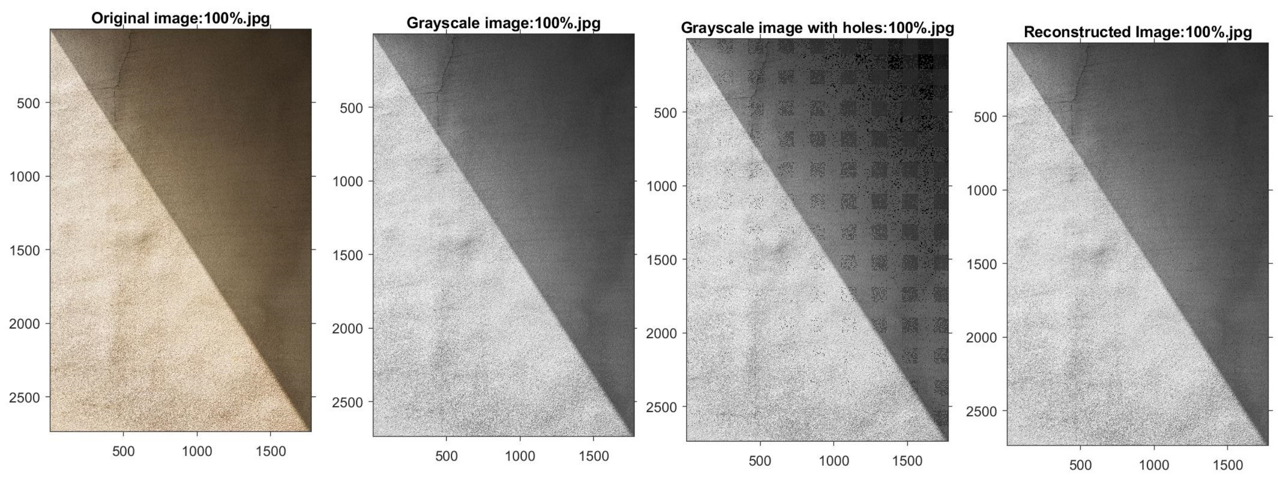


Figure 22: Vino Li texture image at 100% of the original size with 10% error introduction

D1.2 Landscape Images

Table 3: Table showing the compression ratio, PSNR and MSE of the different landscape images at different resolutions

Amanda Kerr				
% Smaller	**Image Size (% of the original image)**	**Compression Ratio**	**PSNR (dB)**	**MSE**
0	100	1.6781	33.005202	32.55069
20	80	1.6423	33.134510	31.59580
40	60	1.5921	33.744086	27.45818
60	40	1.5229	33.850308	26.79474
80	20	1.4058	35.022104	20.45832
Pietro De Grandi				
0	100	1.2193	32.062417	40.44256
20	80	1.2143	31.629805	44.67863
40	60	1.2048	31.845366	42.51515
60	40	1.1897	32.294324	38.33963
80	20	1.1735	30.903258	52.81467
Simon Matzinger				
0	100	1.4106	29.627782	25.294570
20	80	1.4068	28.991301	26.745190
40	60	1.4194	28.404243	29.376720
60	40	1.5082	28.461352	27.316790
80	20	1.0895	28.409550	32.548470

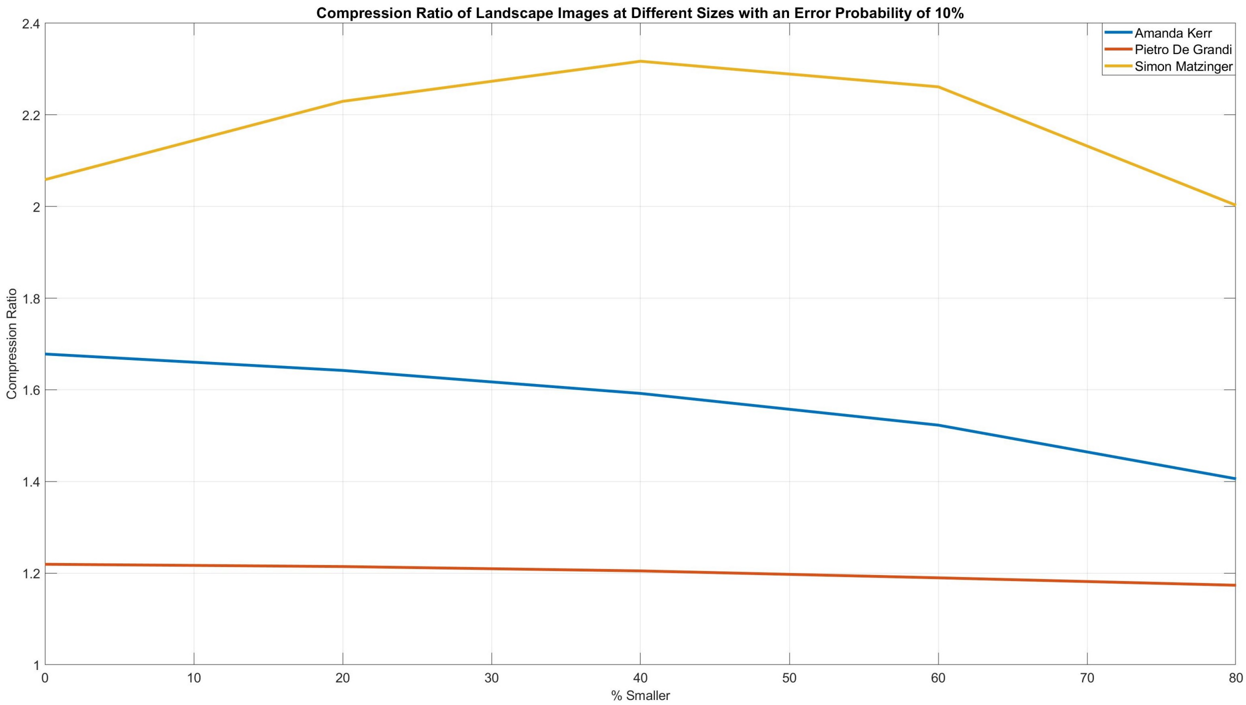


Figure 23: Graph showing the compression ratio against the size of the image for landscape images

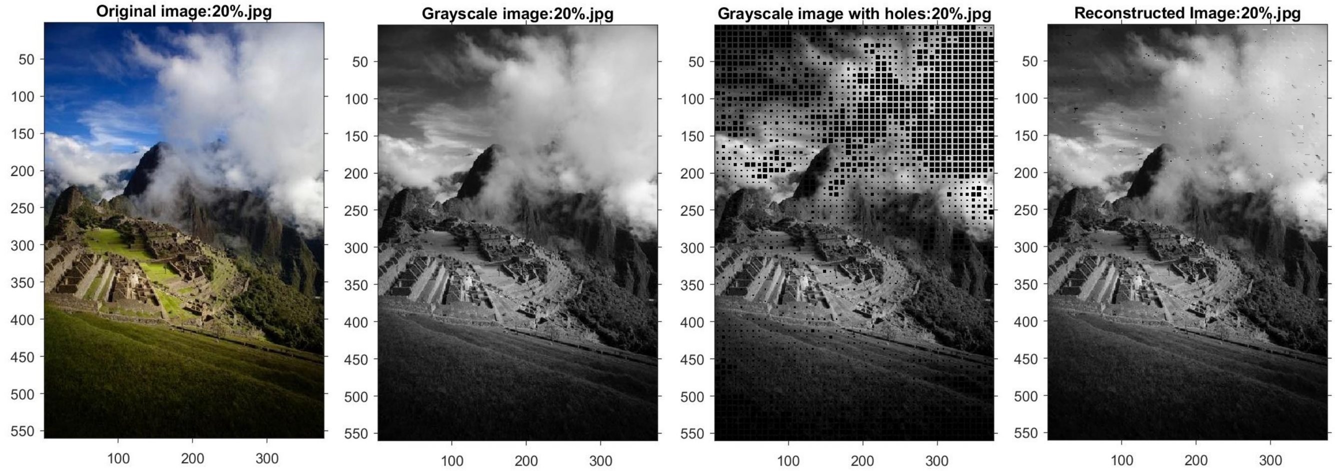


Figure 24: Amanda Kerr landscape image at 20% of the original size with 10% error introduction

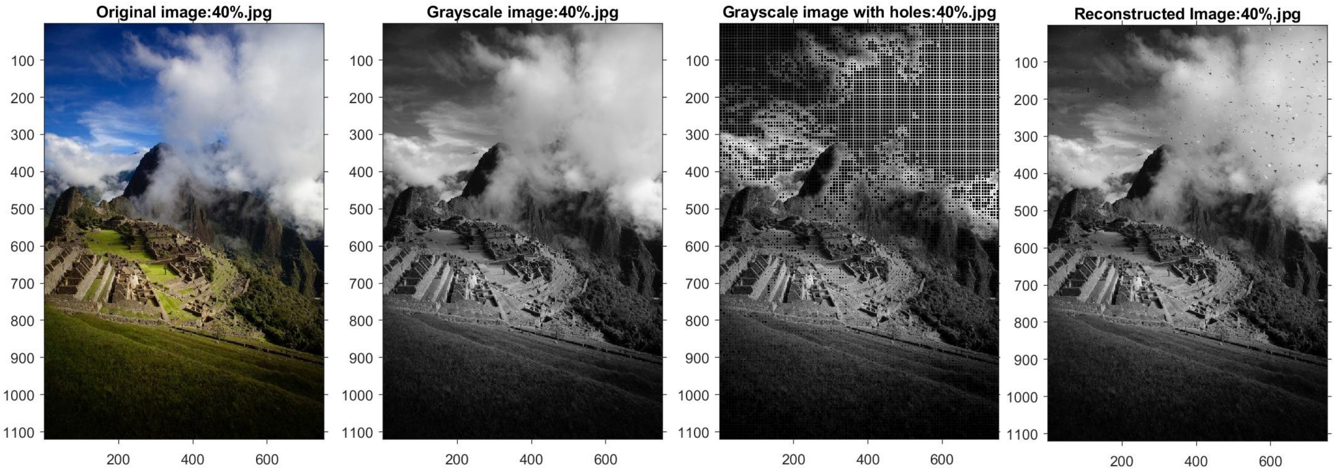


Figure 25: Amanda Kerr landscape image at 40% of the original size with 10% error introduction

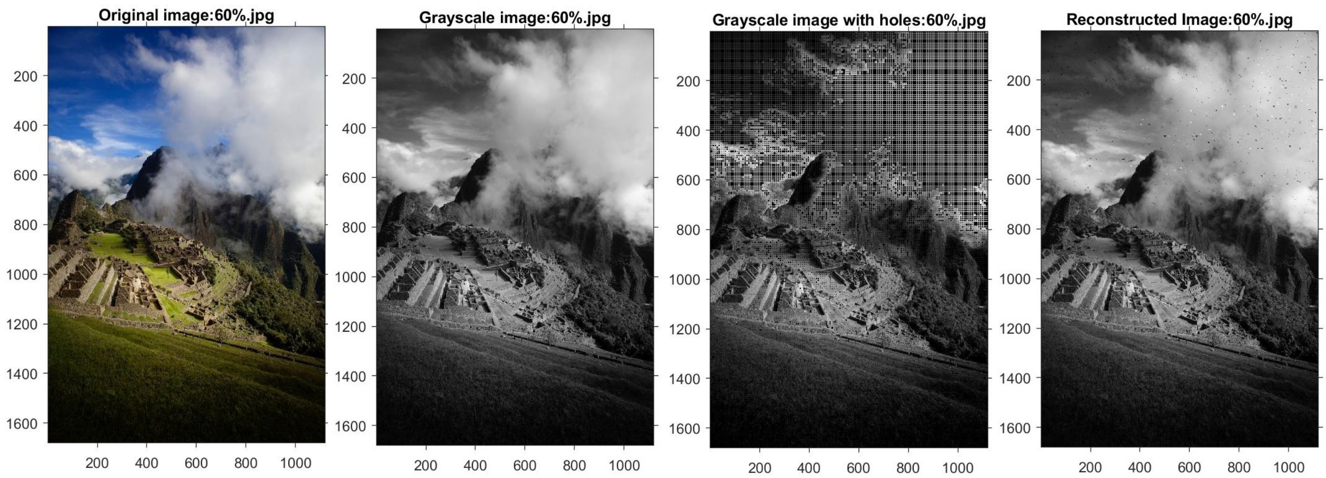


Figure 26: Amanda Kerr landscape image at 60% of the original size with 10% error introduction

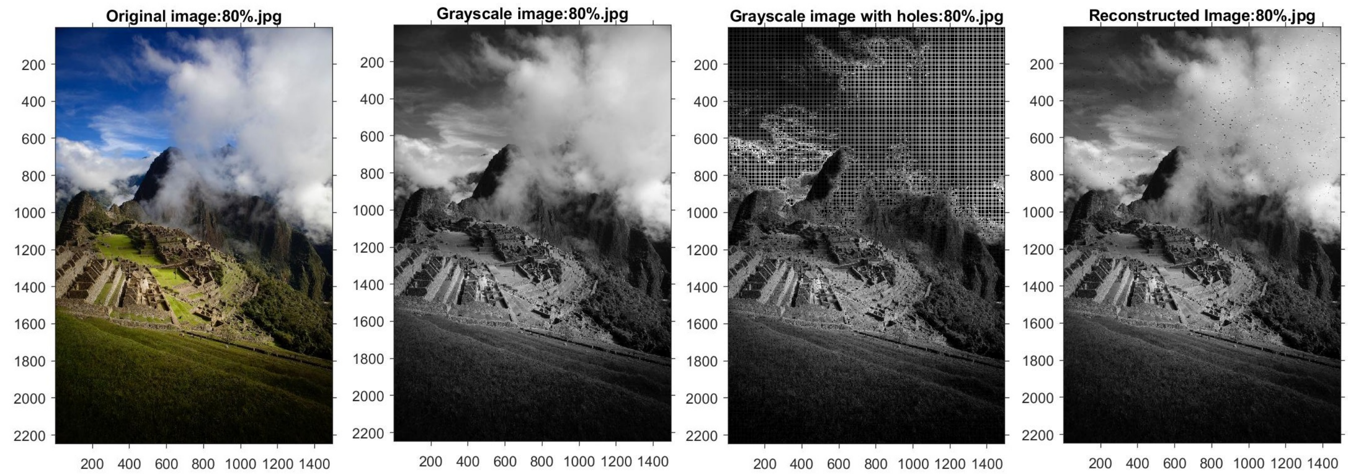


Figure 27: Amanda Kerr landscape image at 80% of the original size with 10% error introduction

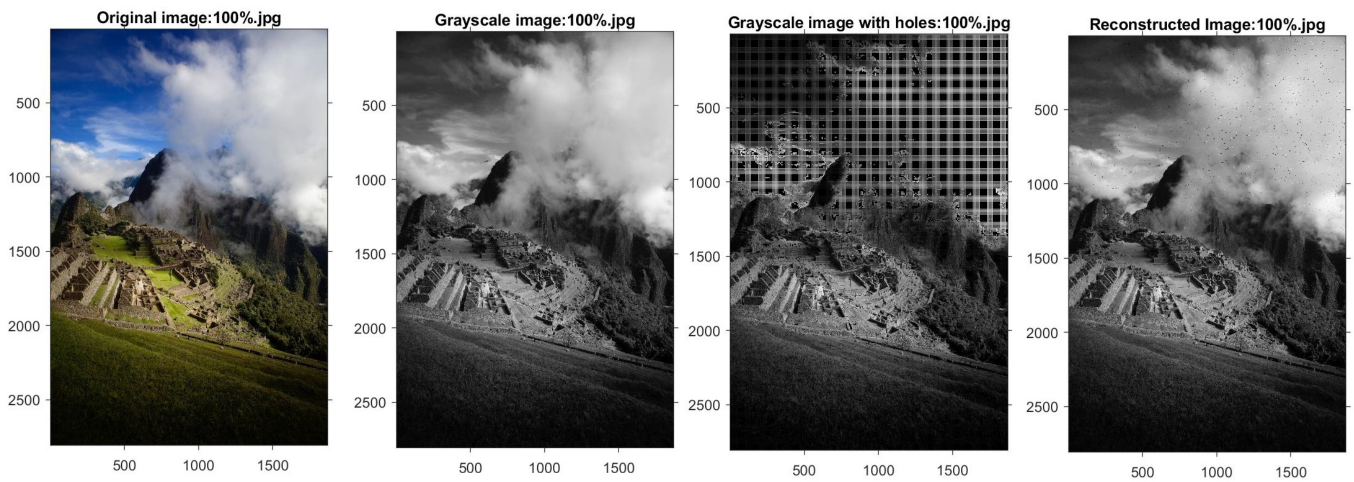


Figure 28: Amanda Kerr landscape image at 100% of the original size with 10% error introduction

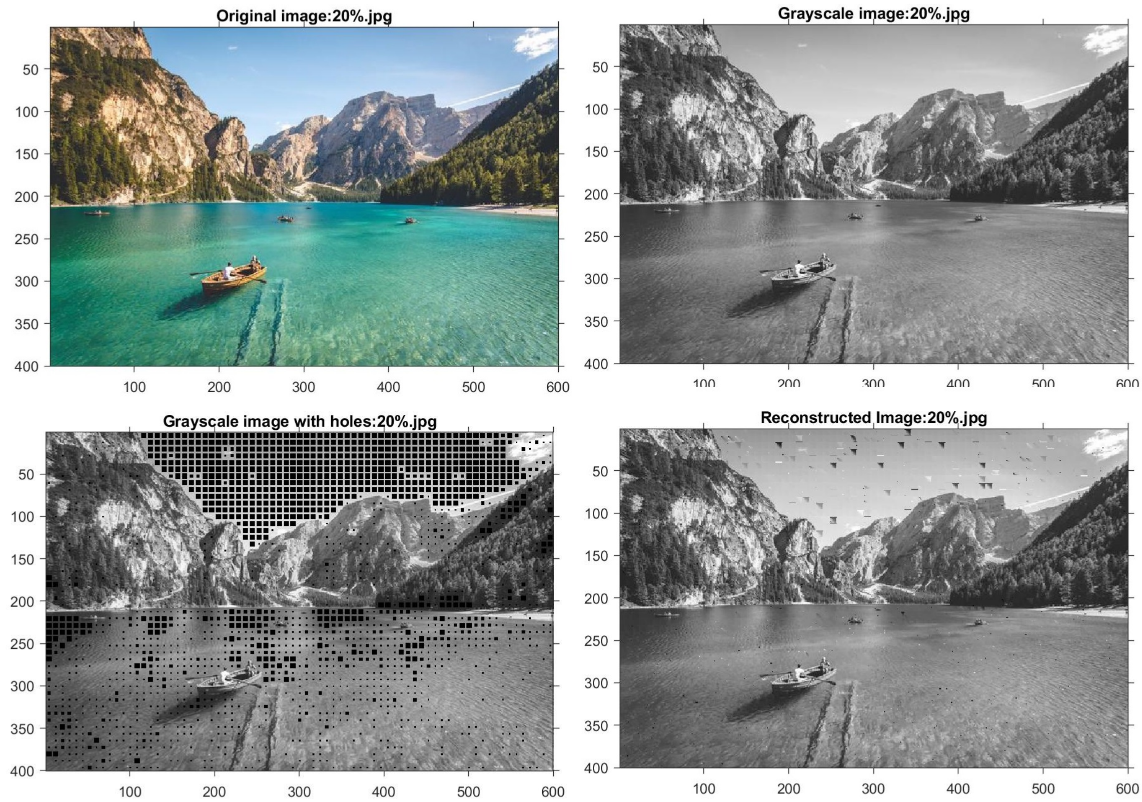


Figure 29: Pietro De Grandi landscape image at 20% of the original size with 10% error introduction

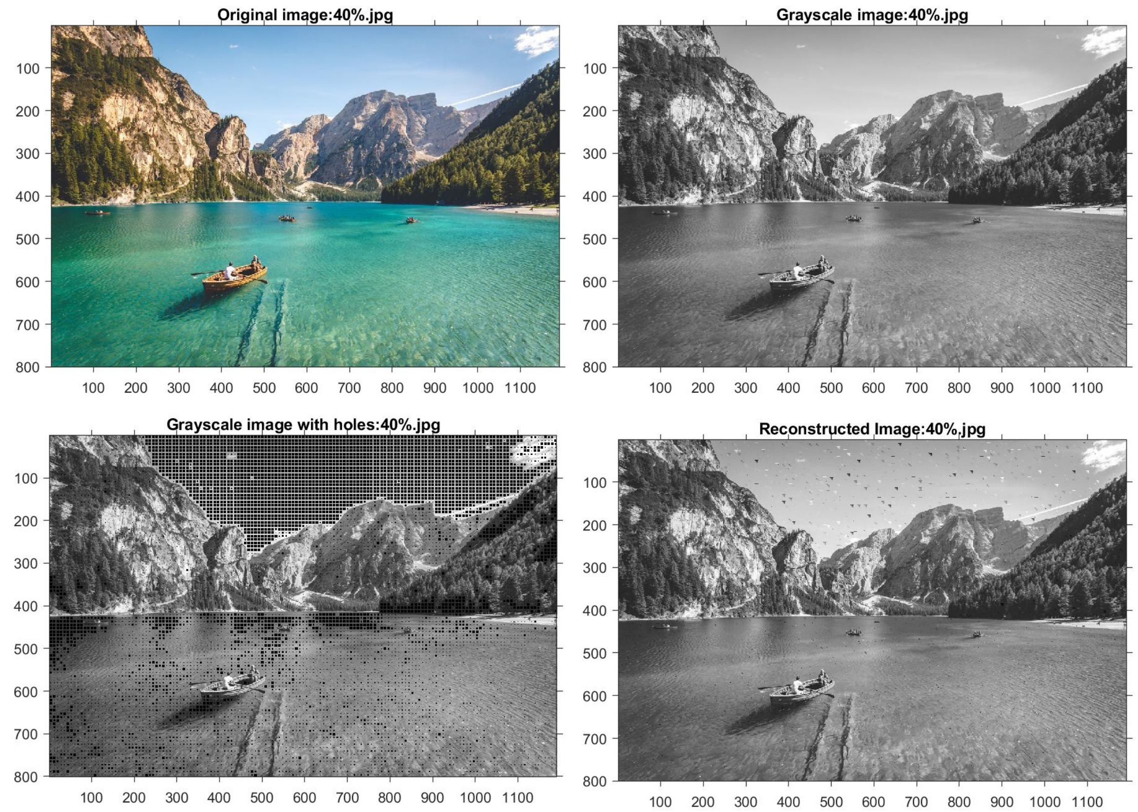


Figure 30: Pietro De Grandi landscape image at 40% of the original size with 10% error introduction

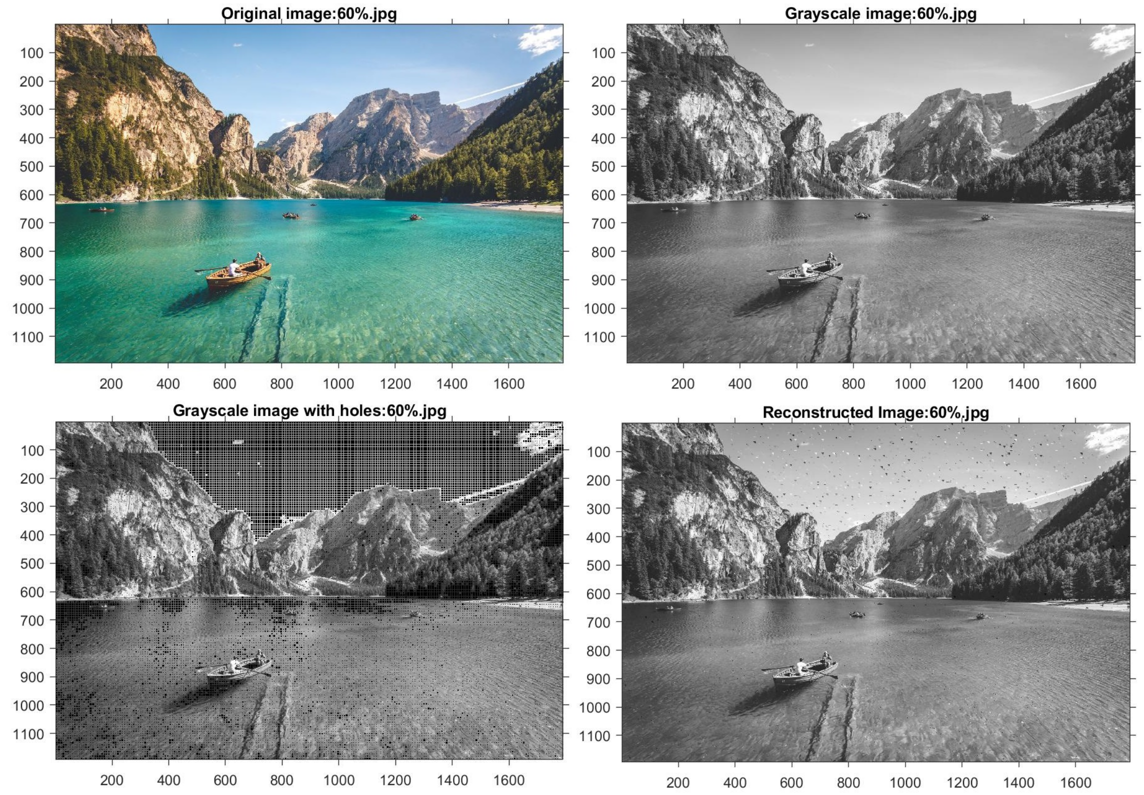


Figure 31: Pietro De Grandi landscape image at 60% of the original size with 10% error introduction

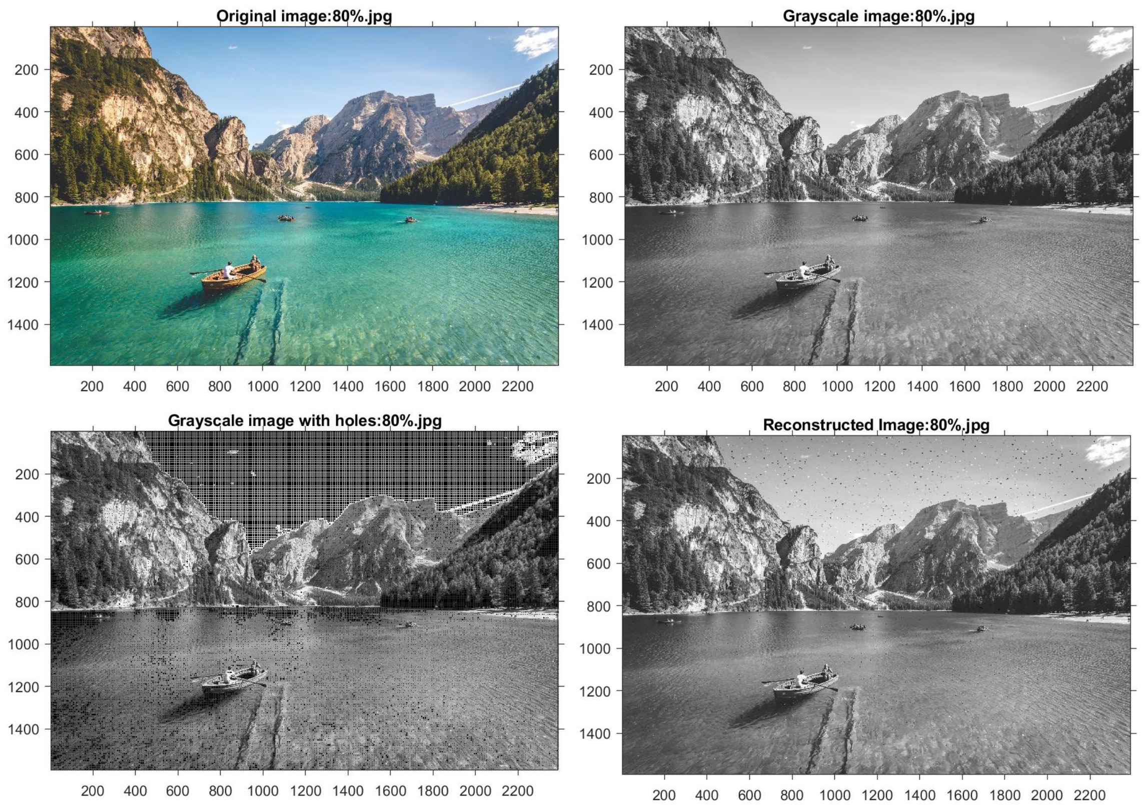


Figure 32: Pietro De Grandi landscape image at 80% of the original size with 10% error introduction

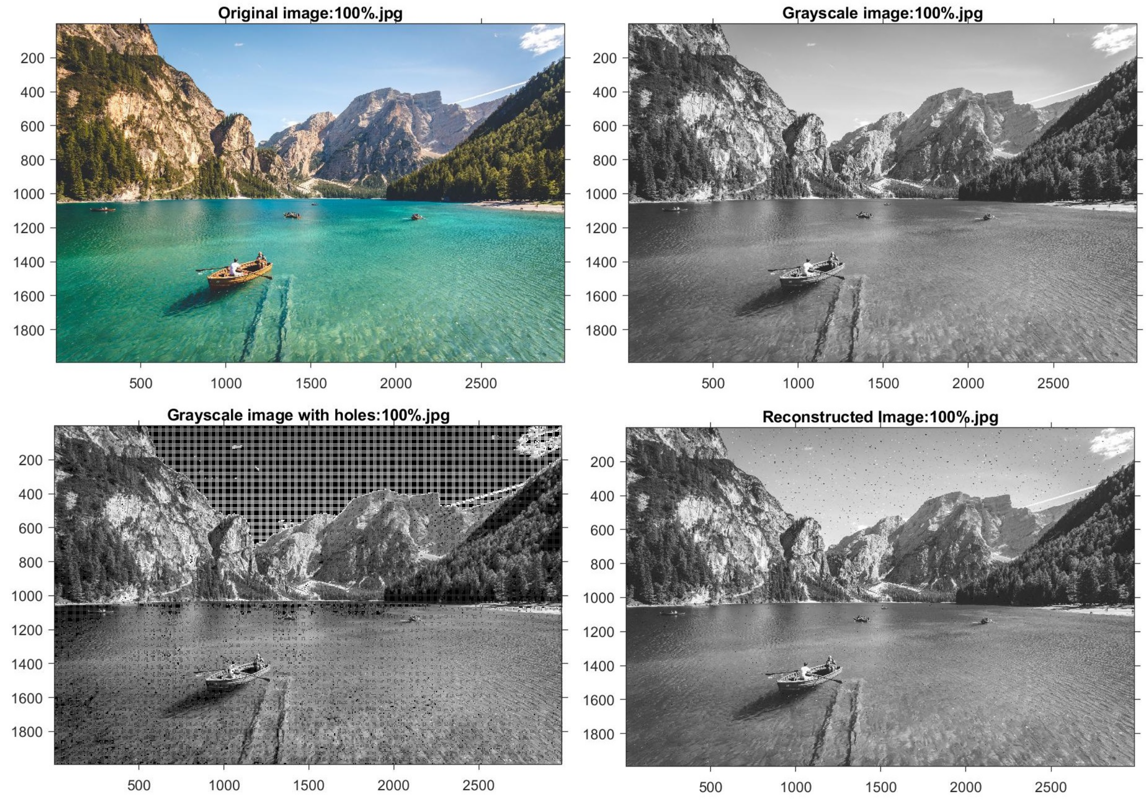


Figure 33: Pietro De Grandi landscape image at 100% of the original size with 10% error introduction

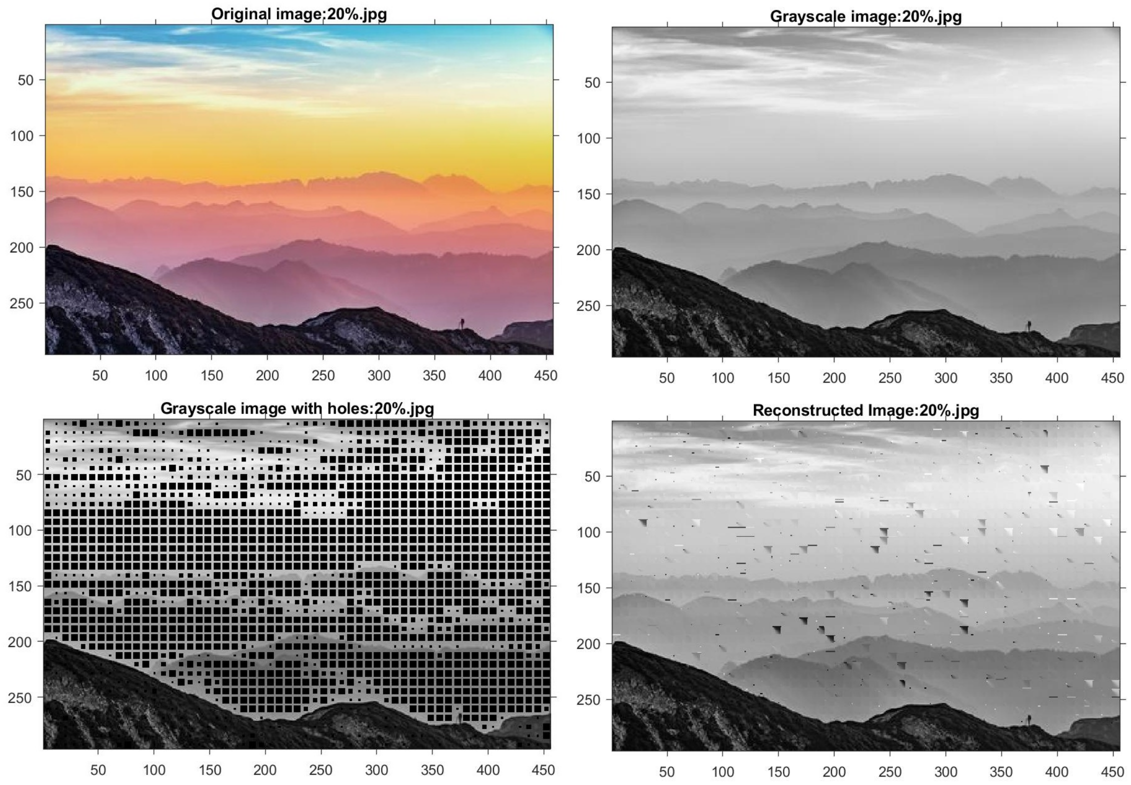


Figure 34: Simon Matzinger landscape image at 20% of the original size with 10% error introduction

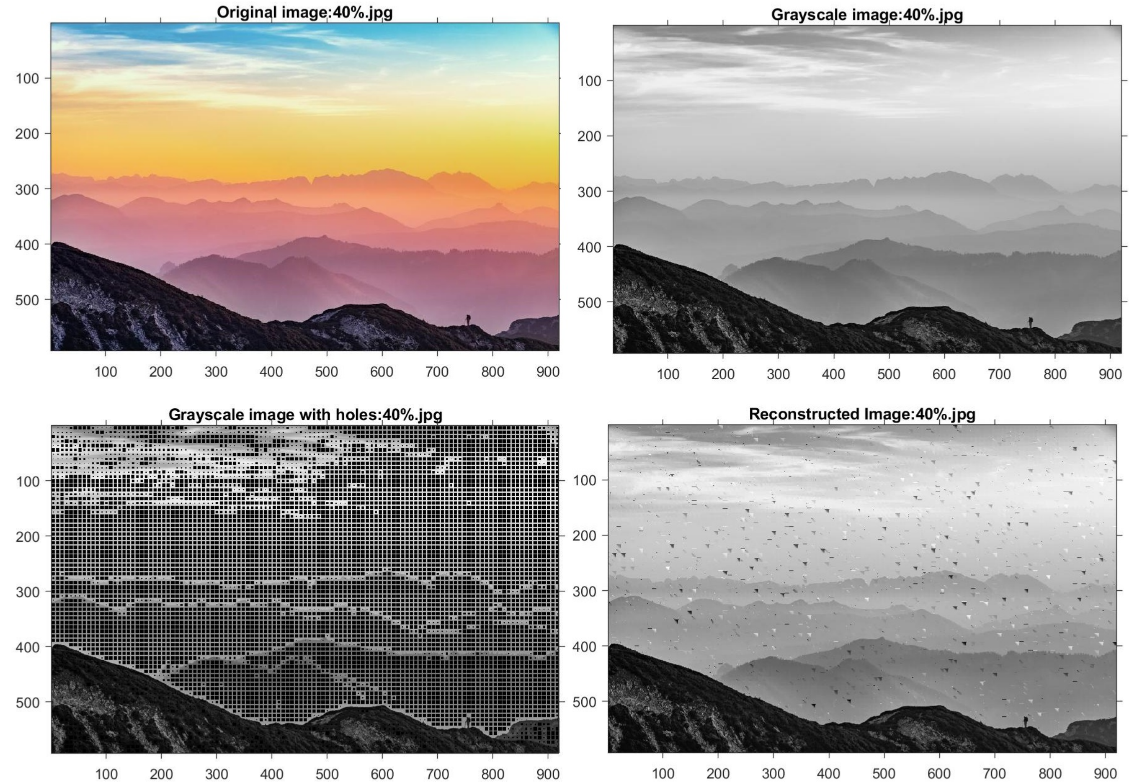


Figure 35: Simon Matzinger landscape image at 40% of the original size with 10% error introduction

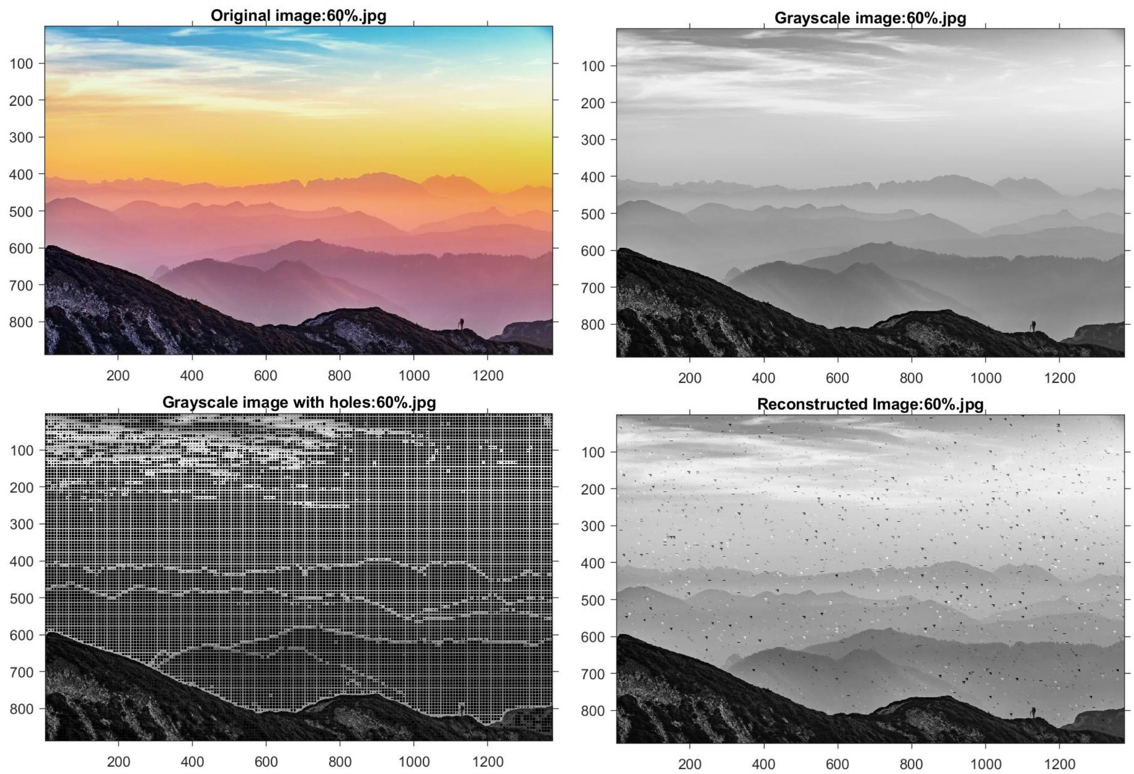


Figure 36: Simon Matzinger landscape image at 60% of the original size with 10% error introduction

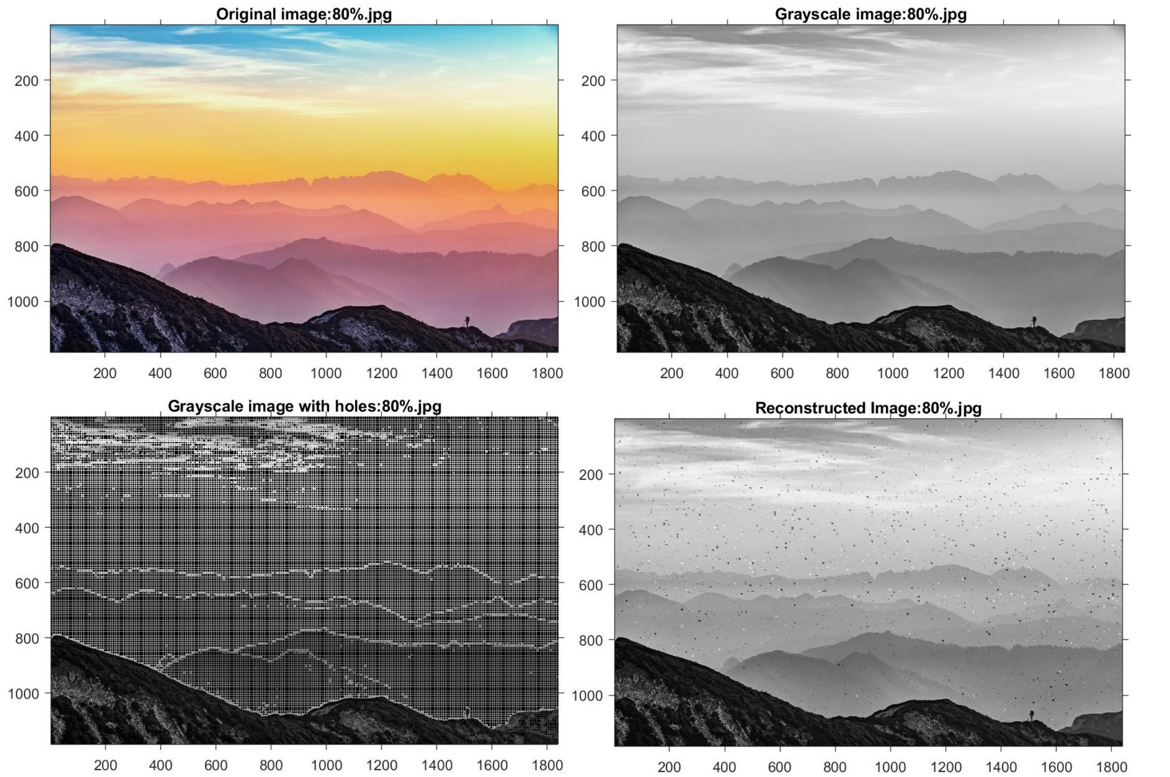


Figure 37: Simon Matzinger landscape image at 80% of the original size with 10% error introduction

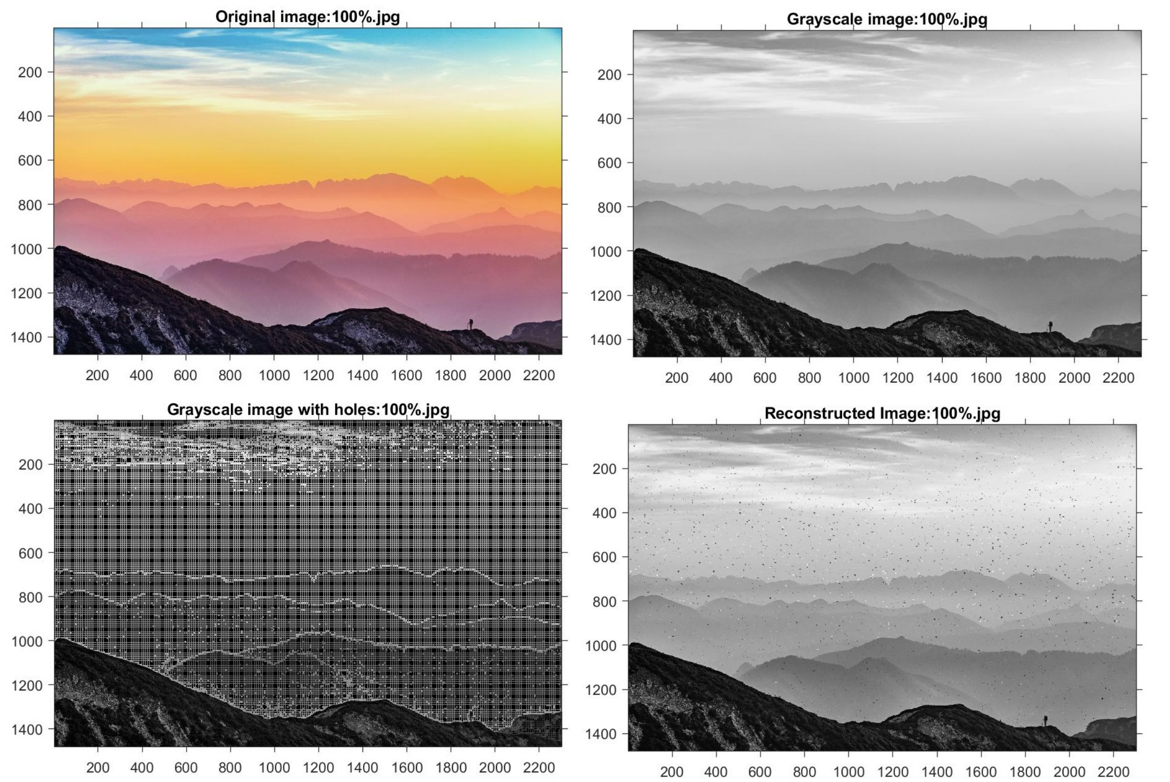


Figure 38: Simon Matzinger landscape image at 100% of the original size with 10% error introduction

D1.3 High Contrast Images

Table 4: Table showing the compression ratio, PSNR and MSE of the different high contrast images at different resolutions

Ananth Pai				
% Smaller	**Image Size (% of the original image)**	**Compression Ratio**	**PSNR (dB)**	**MSE**
0	100	1.9715	35.589593	17.95234
20	80	1.8987	35.909049	16.67921
40	60	1.8115	35.856971	16.88042
60	40	1.6956	36.317748	15.18118
80	20	1.5463	36.378931	14.96881
Ricardo				
0	100	2.6827	27.006936	129.53490
20	80	2.6637	27.003949	129.62400
40	60	2.6217	26.936463	131.65400
60	40	2.5352	26.962416	130.86960
80	20	2.4577	26.709159	138.72810
Yerko Lucic				
0	100	1.9551	34.587437	22.611880
20	80	1.9846	34.043248	25.630410
40	60	1.9883	34.128872	25.130040
60	40	1.9022	33.636146	28.149180
80	20	1.6492	35.079899	20.187870

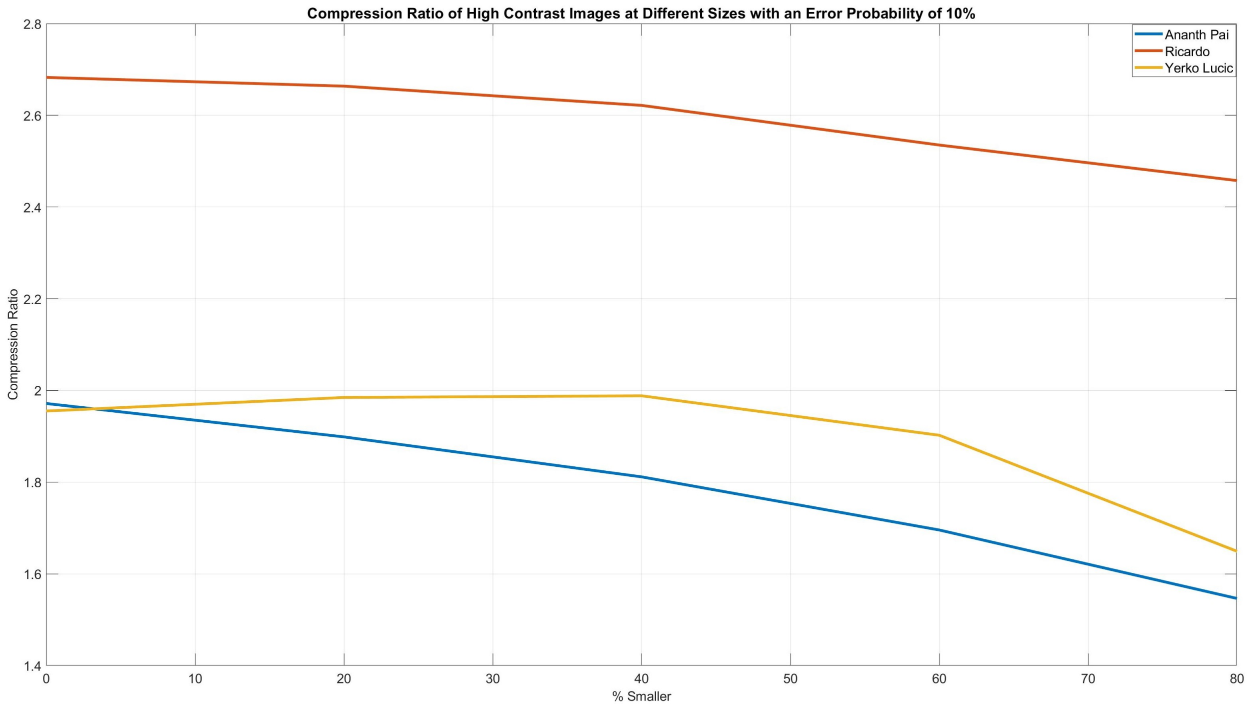


Figure 39: Graph showing the compression ratio against the size of the image for high contrast images

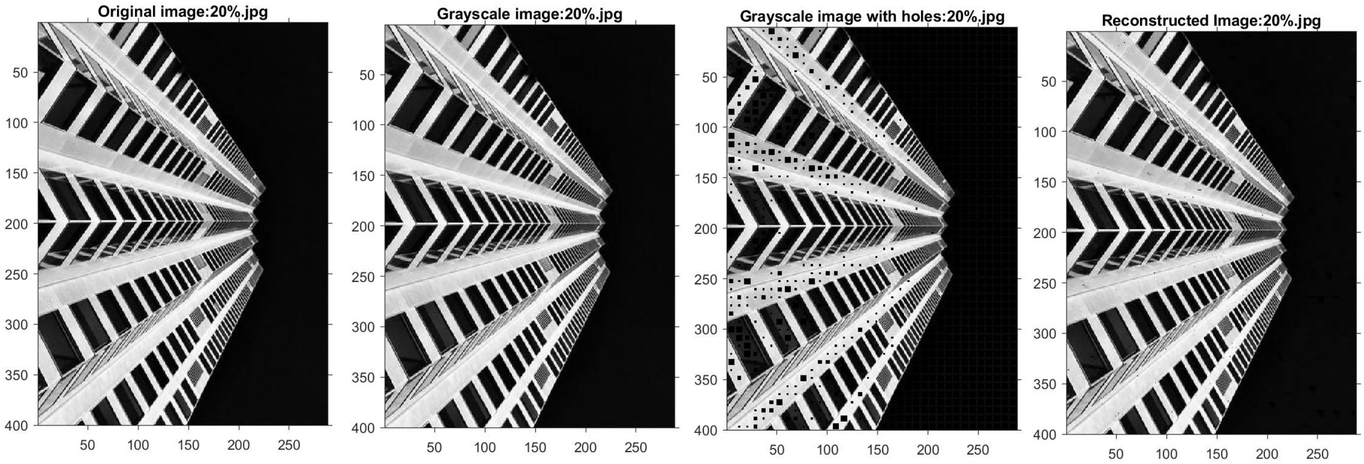


Figure 40: Ananth Pai high contrast image at 20% of the original size with 10% error introduction

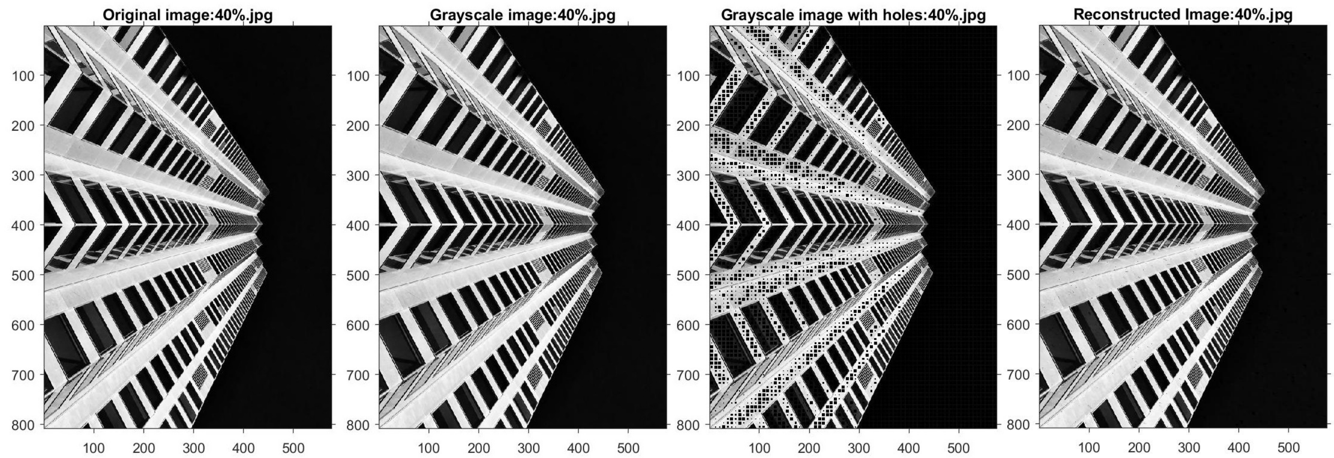


Figure 41: Ananth Pai high contrast image at 40% of the original size with 10% error introduction

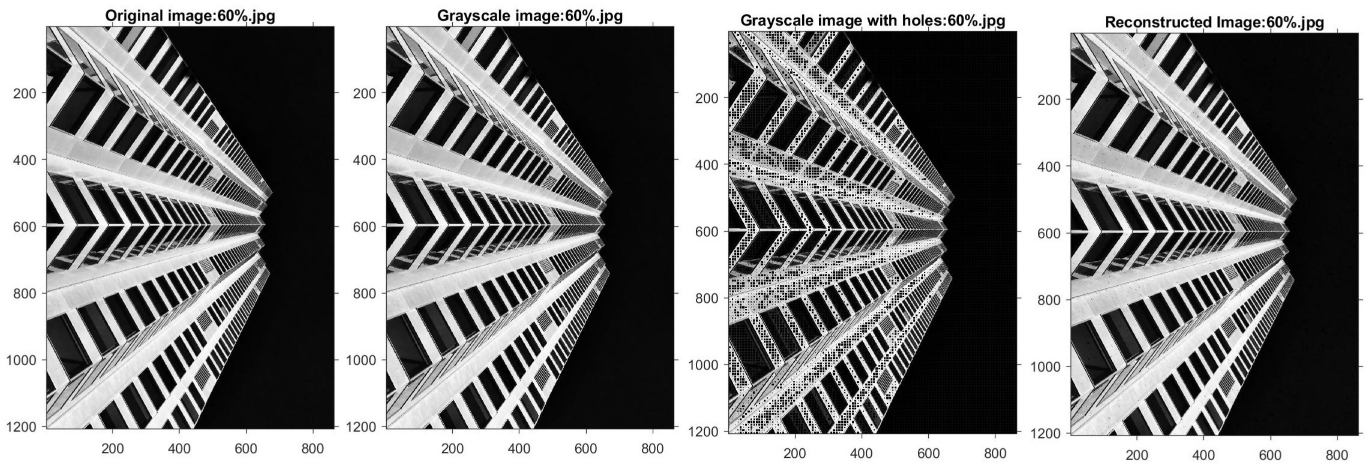


Figure 42: Ananth Pai high contrast image at 60% of the original size with 10% error introduction

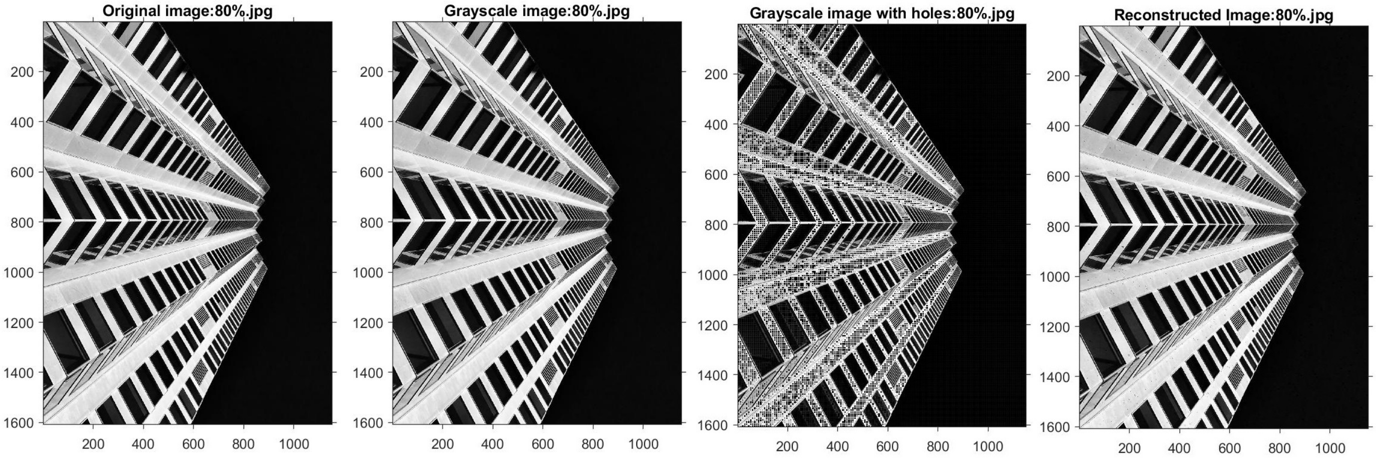


Figure 43: Ananth Pai high contrast image at 80% of the original size with 10% error introduction

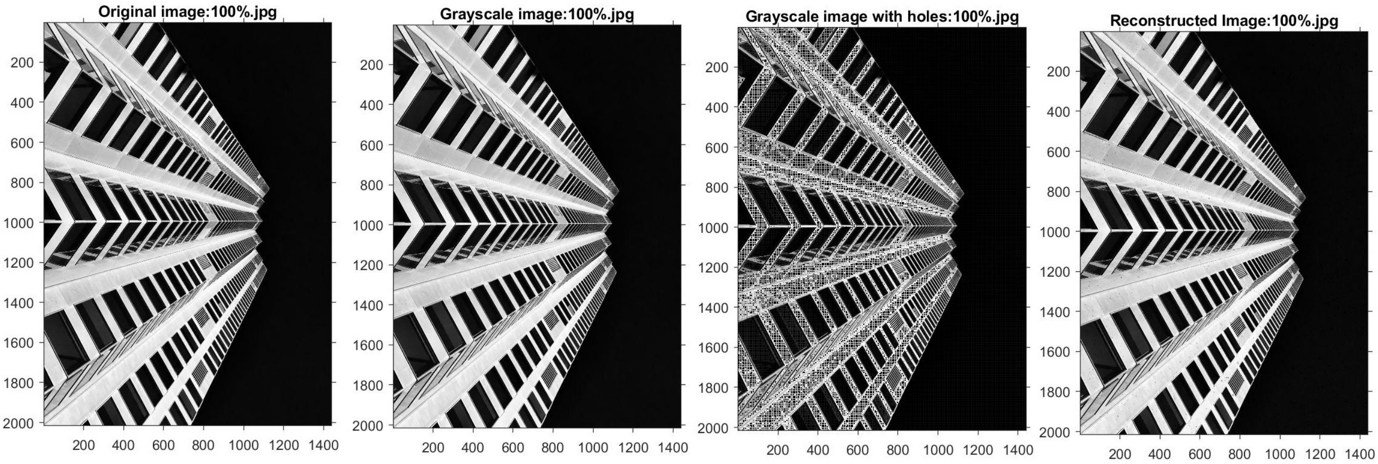


Figure 44: Ananth Pai high contrast image at 100% of the original size with 10% error introduction

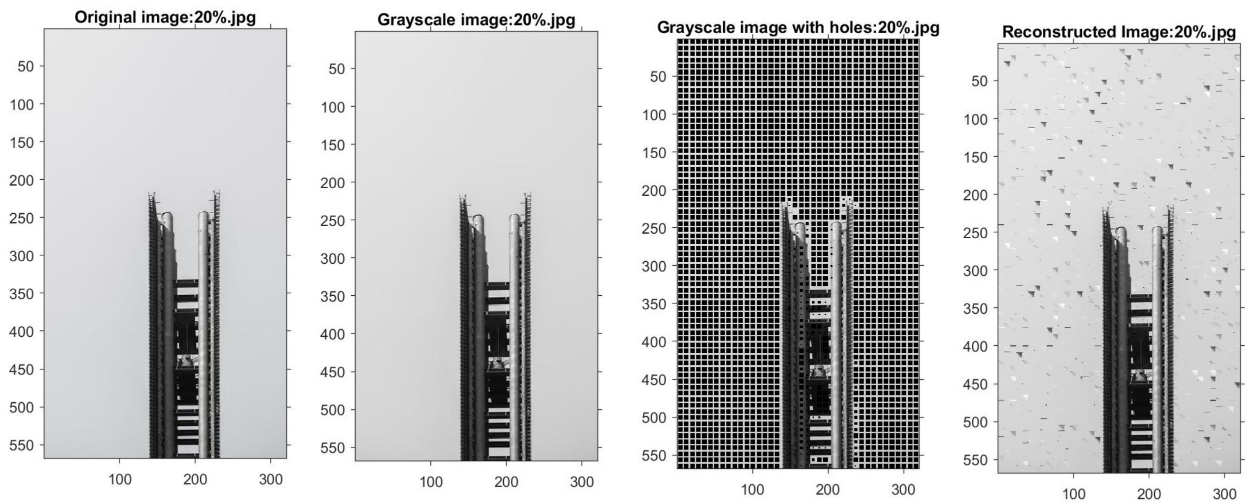


Figure 45: Ricardo high contrast image at 20% of the original size with 10% error introduction

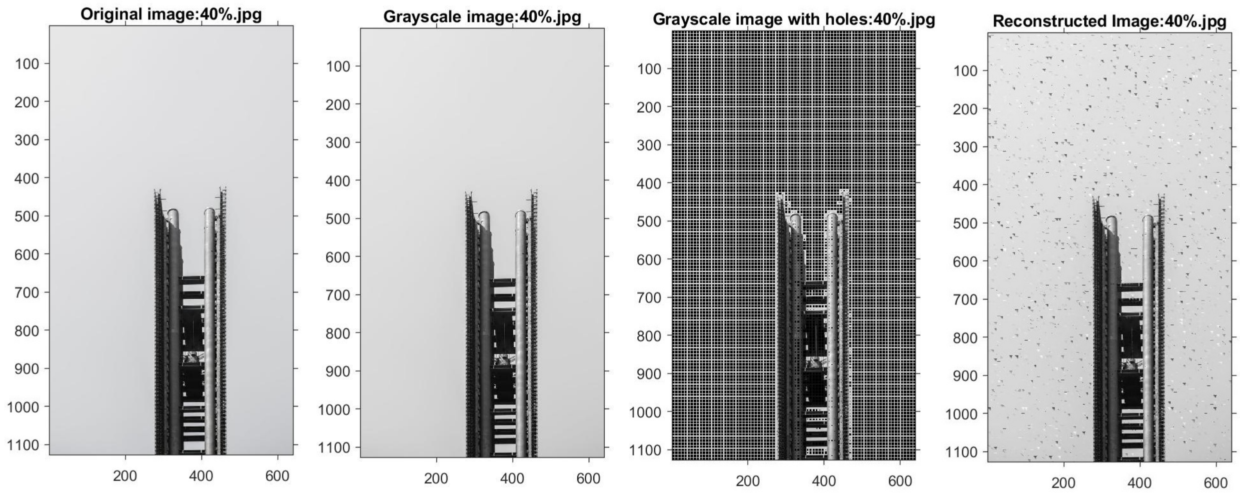


Figure 46: Ricardo high contrast image at 40% of the original size with 10% error introduction

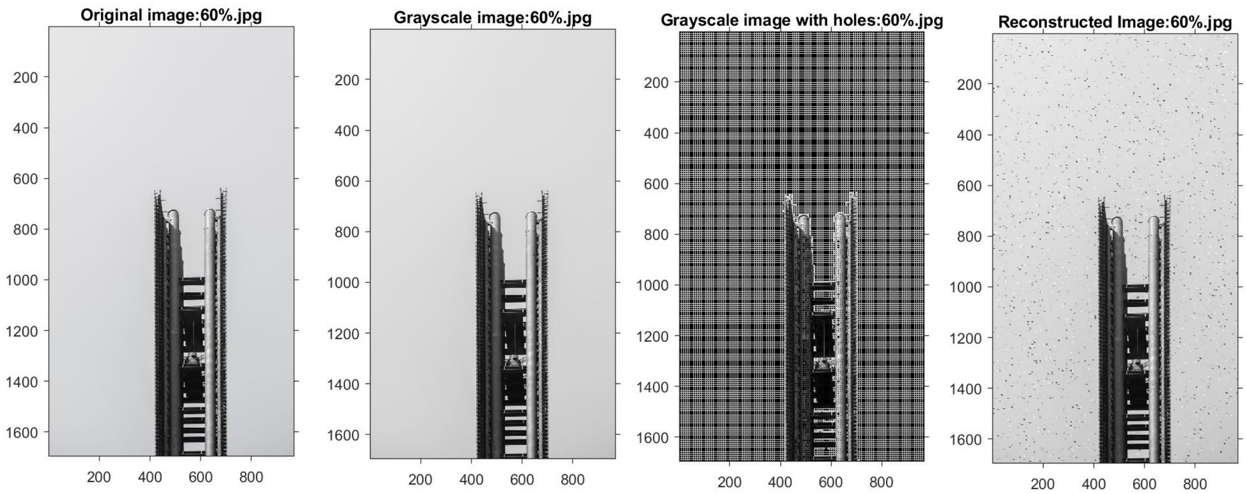


Figure 47: Ricardo high contrast image at 60% of the original size with 10% error introduction

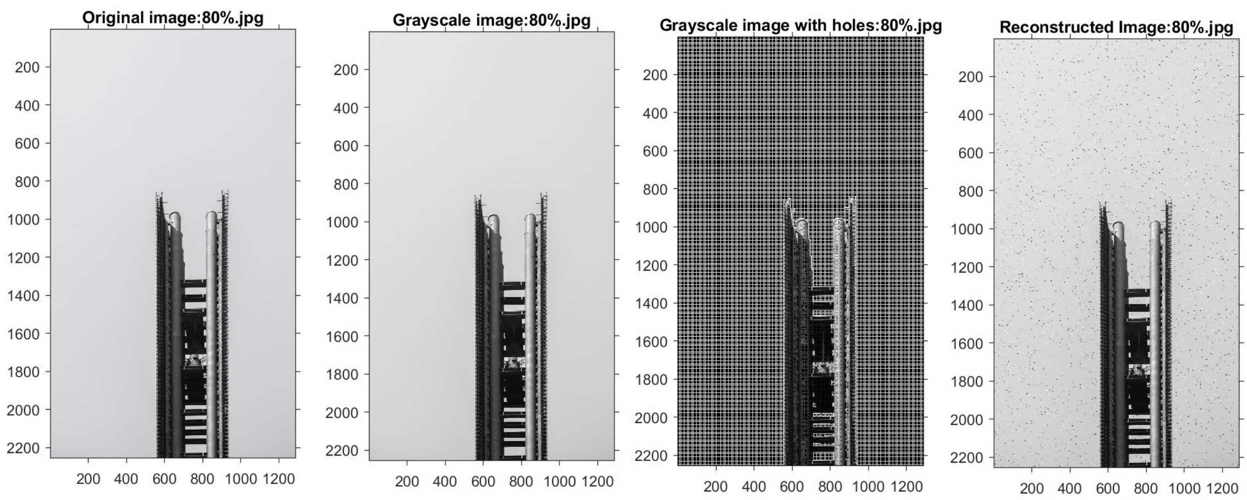


Figure 48: Ricardo high contrast image at 80% of the original size with 10% error introduction

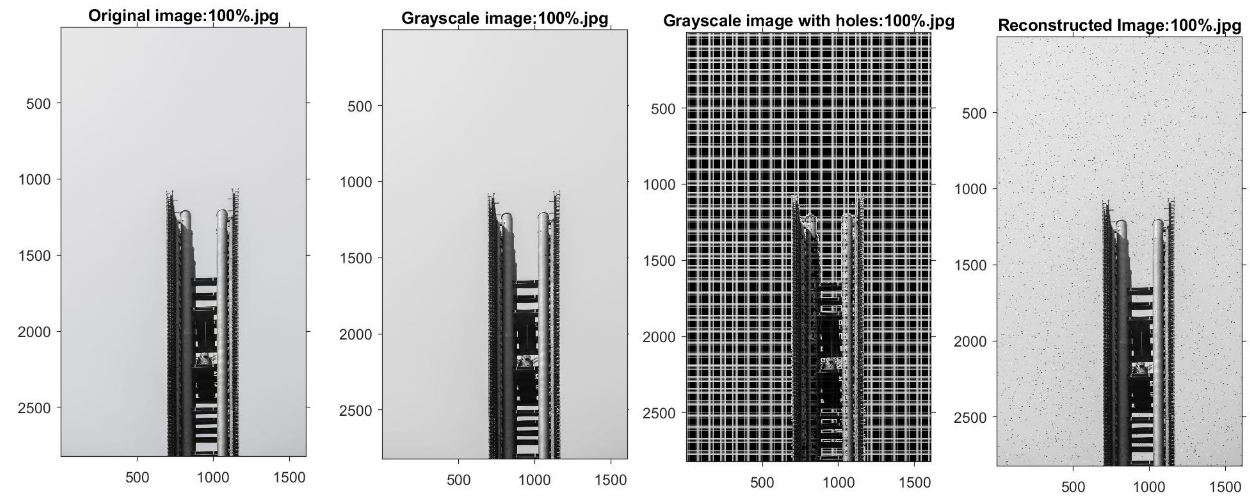


Figure 49: Ricardo high contrast image at 100% of the original size with 10% error introduction

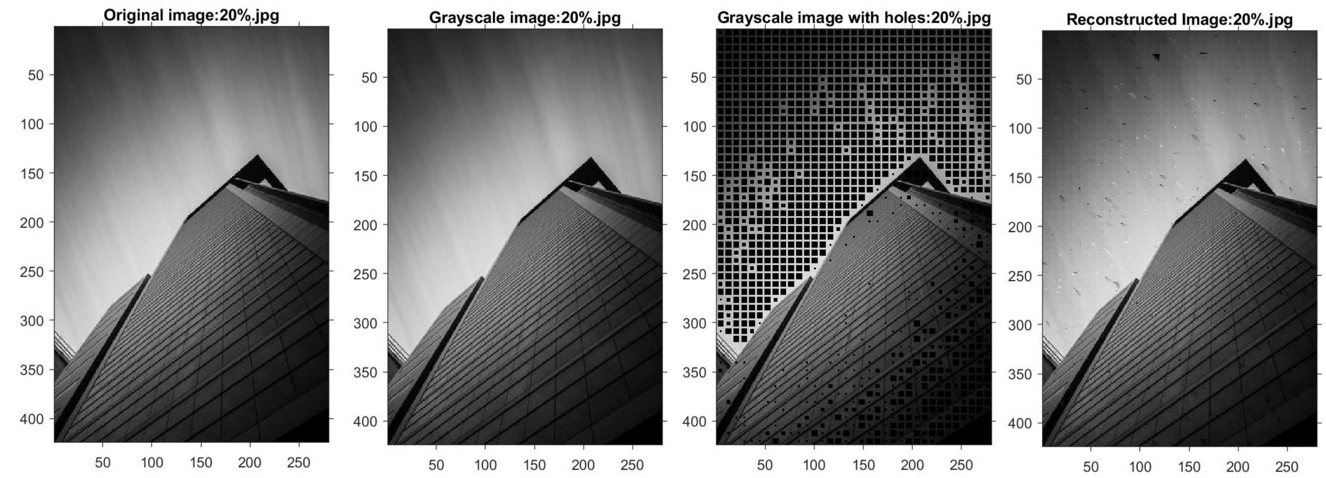


Figure 50: Yerko Lucic high contrast image at 20% of the original size with 10% error introduction

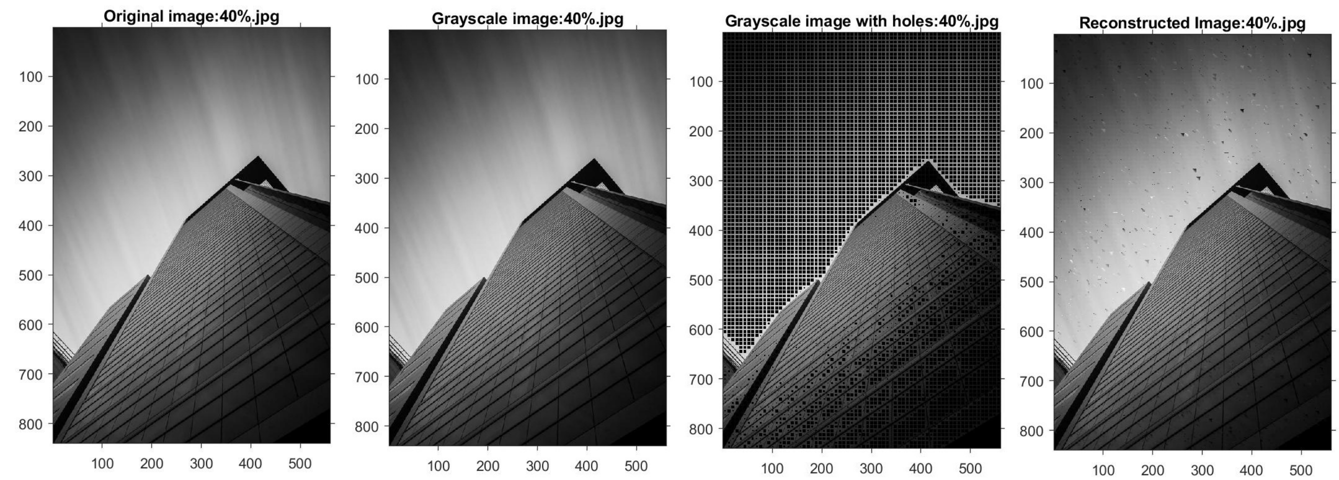


Figure 51: Yerko Lucic high contrast image at 40% of the original size with 10% error introduction

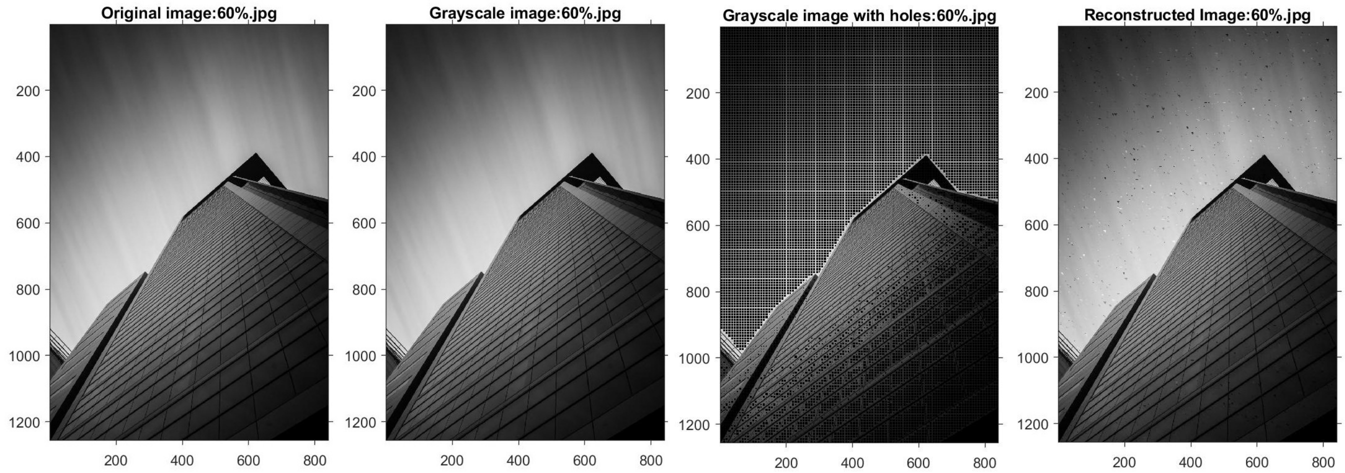


Figure 52: Yerko Lucic high contrast image at 60% of the original size with 10% error introduction

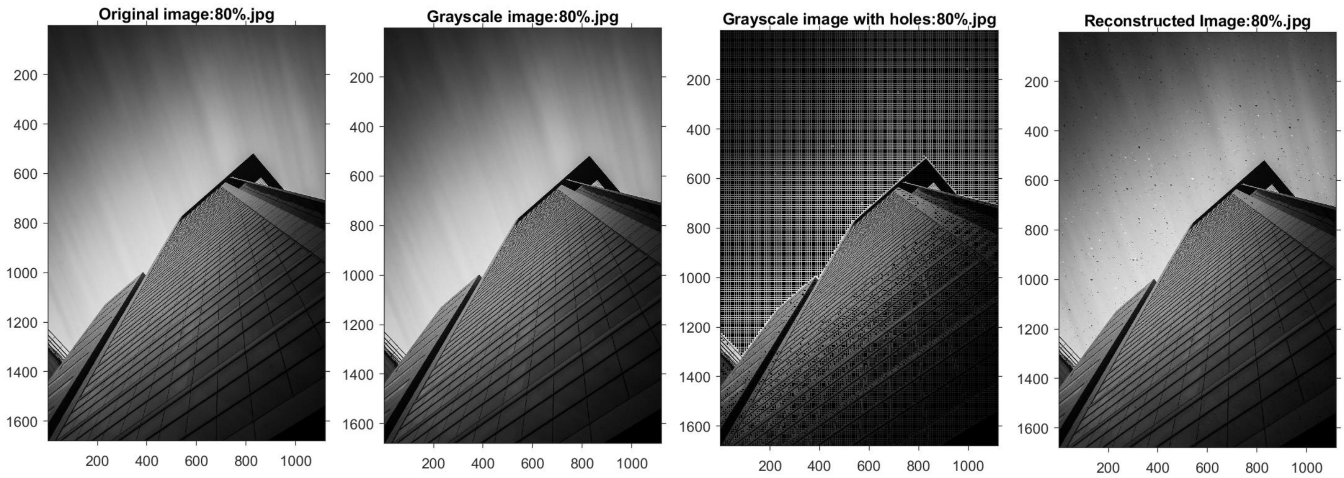


Figure 53: Yerko Lucic high contrast image at 80% of the original size with 10% error introduction

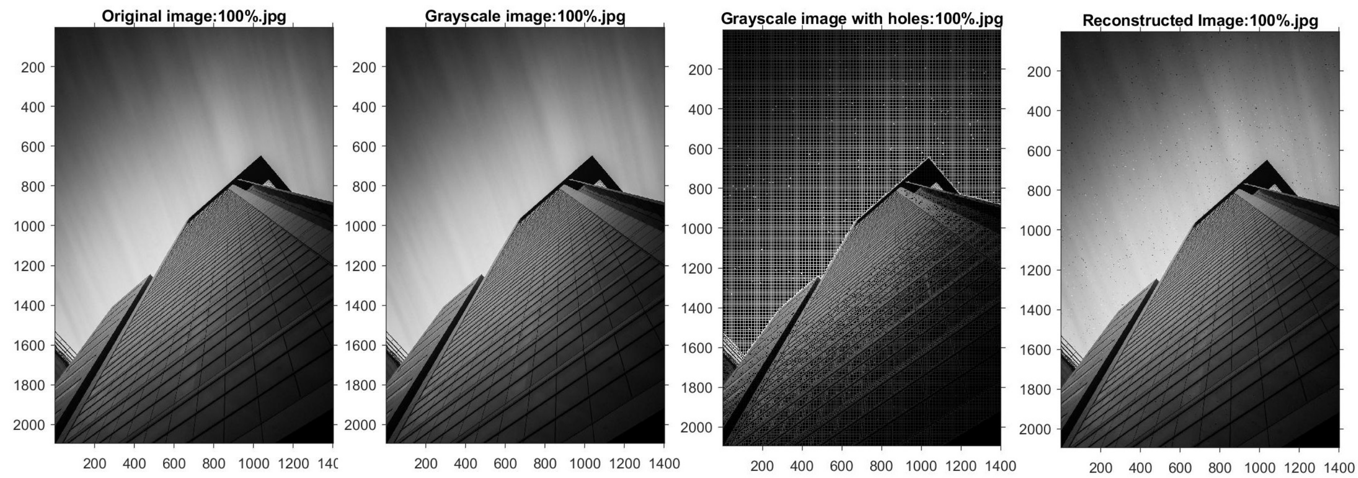


Figure 54: Yerko Lucic high contrast image at 100% of the original size with 10% error introduction

D2 Test Case 2: Average Compression Ratio of the Different Image Types

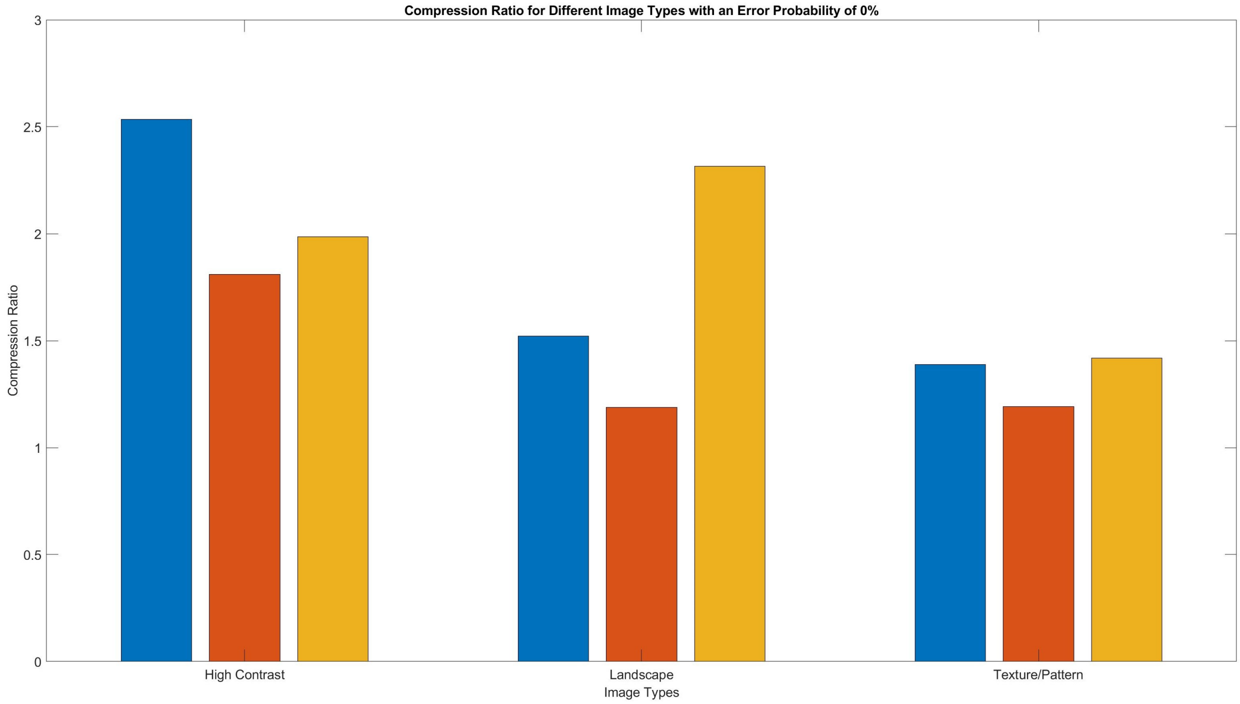


Figure 55: Graph showing the compression ratios of the different types of images

D2.1 Pattern Images

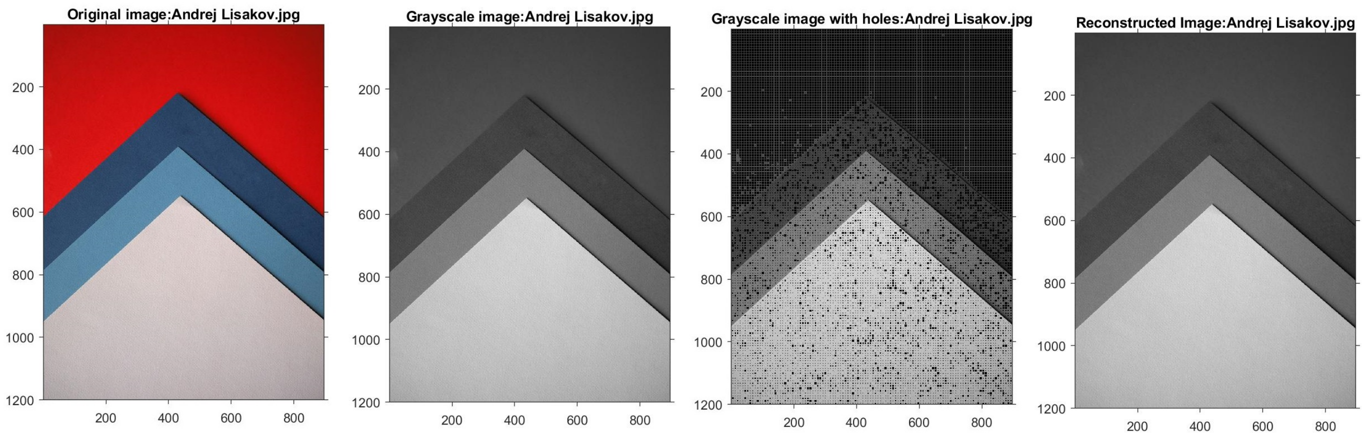


Figure 56: Andrej Lisakov pattern image with 0% error introduction

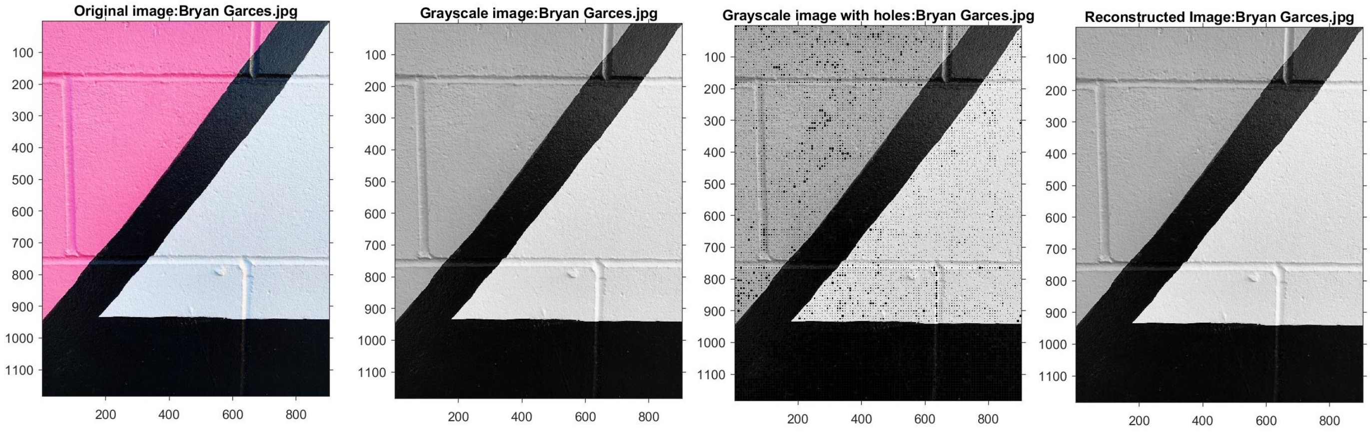


Figure 57: Bryan Garces pattern image with 0% error introduction

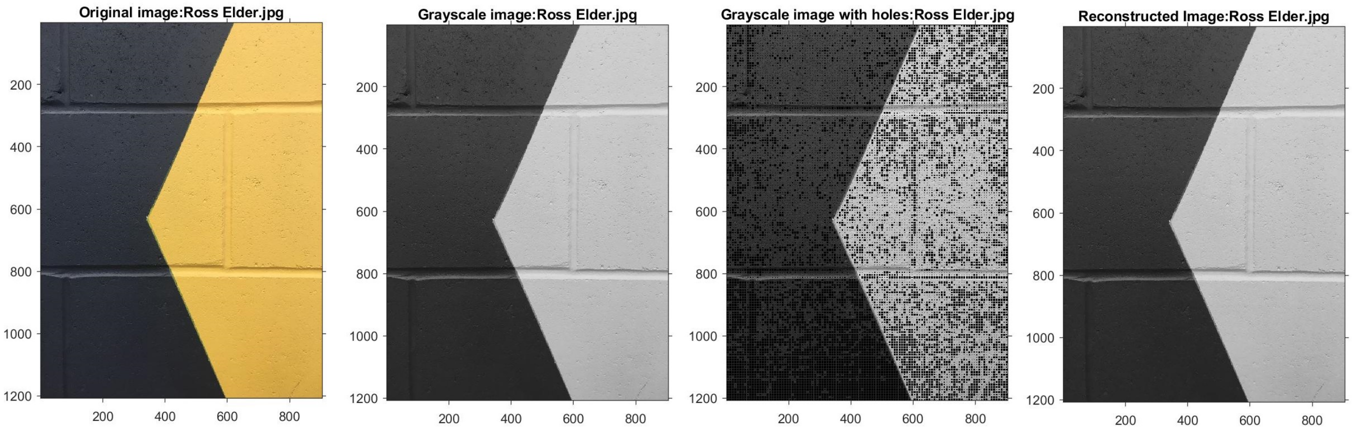


Figure 58: Ross Elder pattern image with 0% error introduction

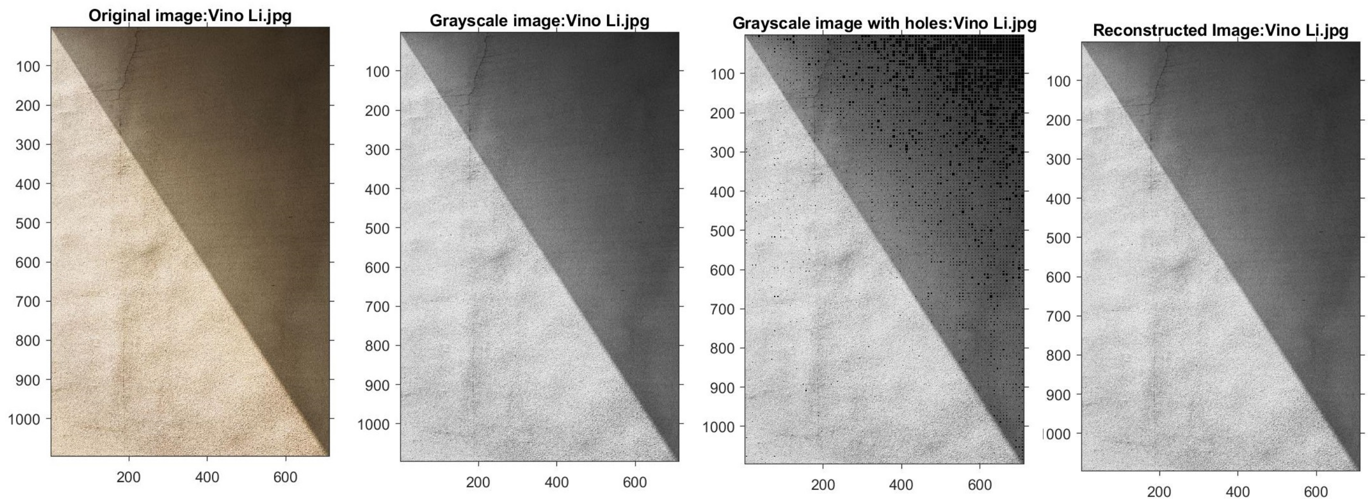


Figure 59: Vino Li pattern image with 0% error introduction

D2.2 Landscape Images

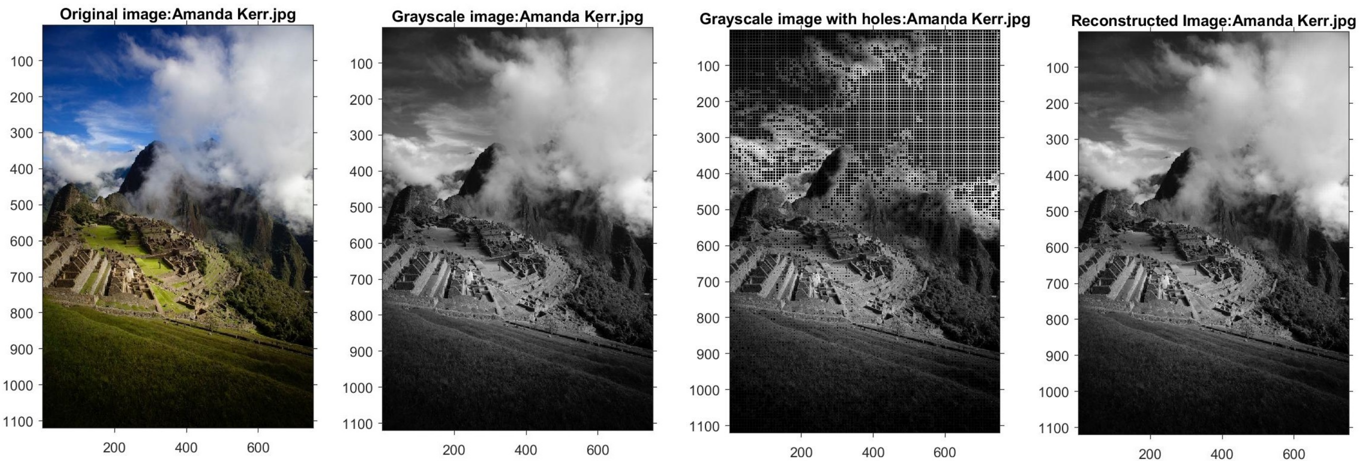


Figure 60: Amanda Kerr landscape image with 0% error introduction

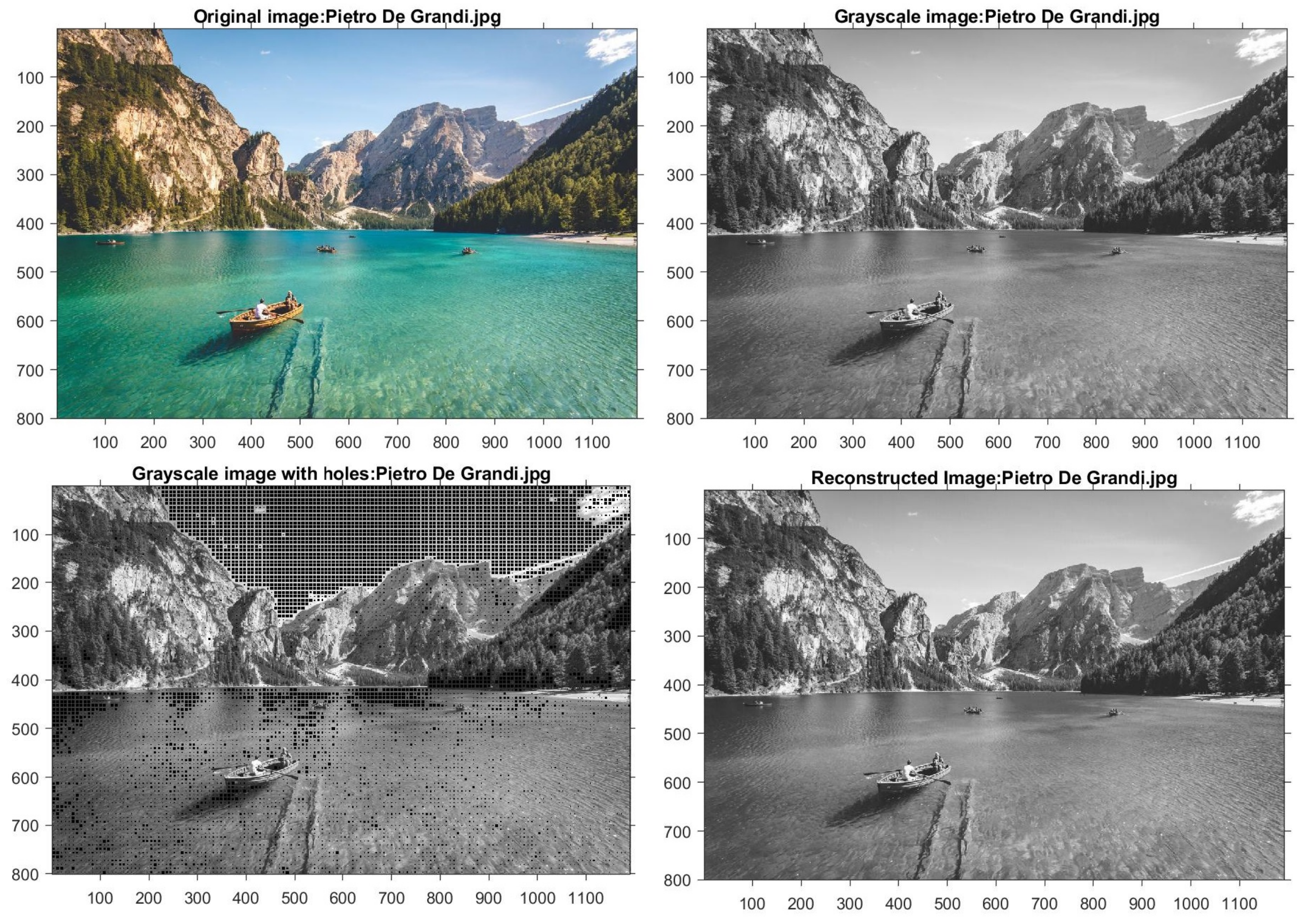


Figure 61: Pietro Di Grandi landscape image with 0% error introduction

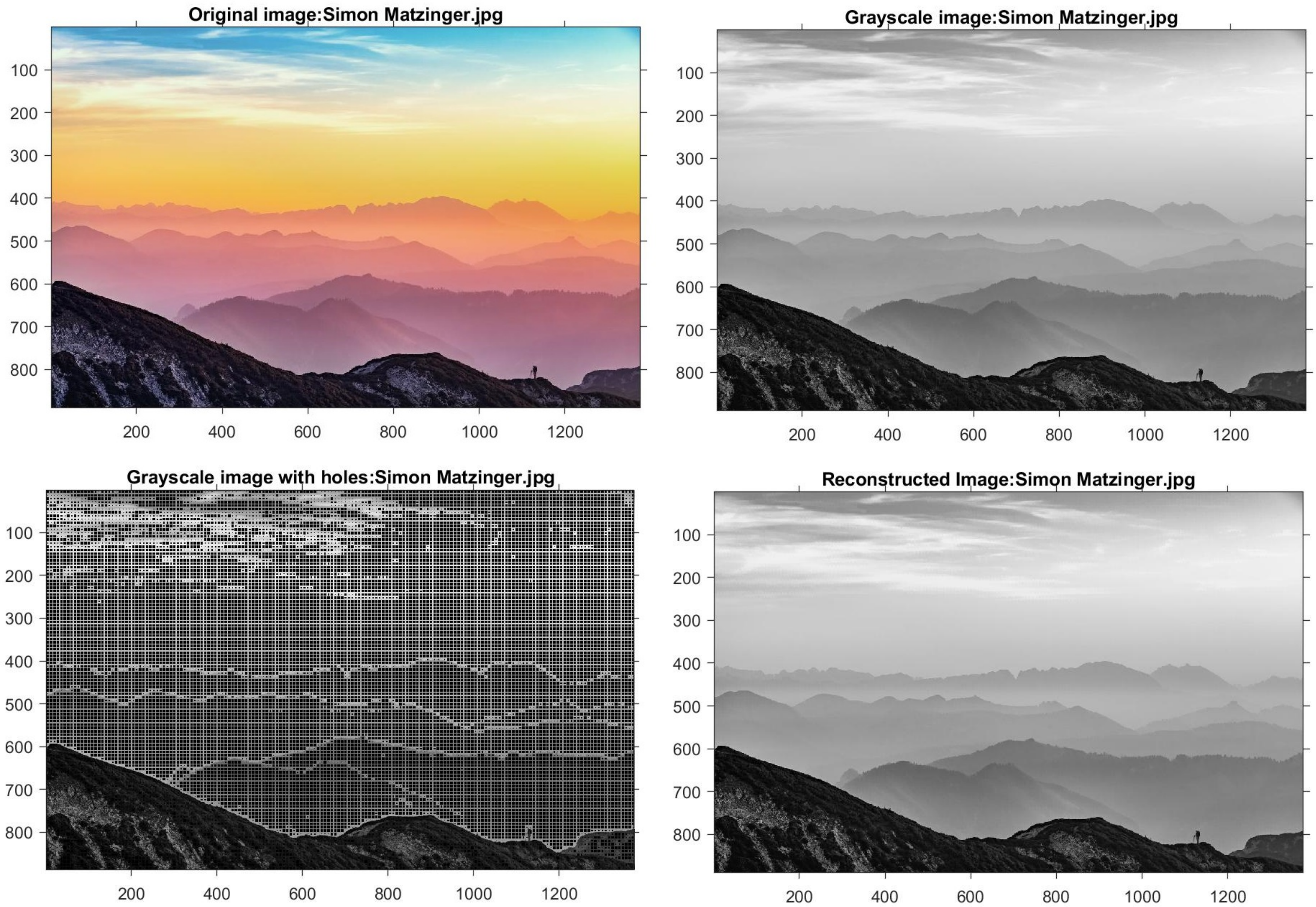


Figure 62: Simon Matzinger landscape image with 0% error introduction

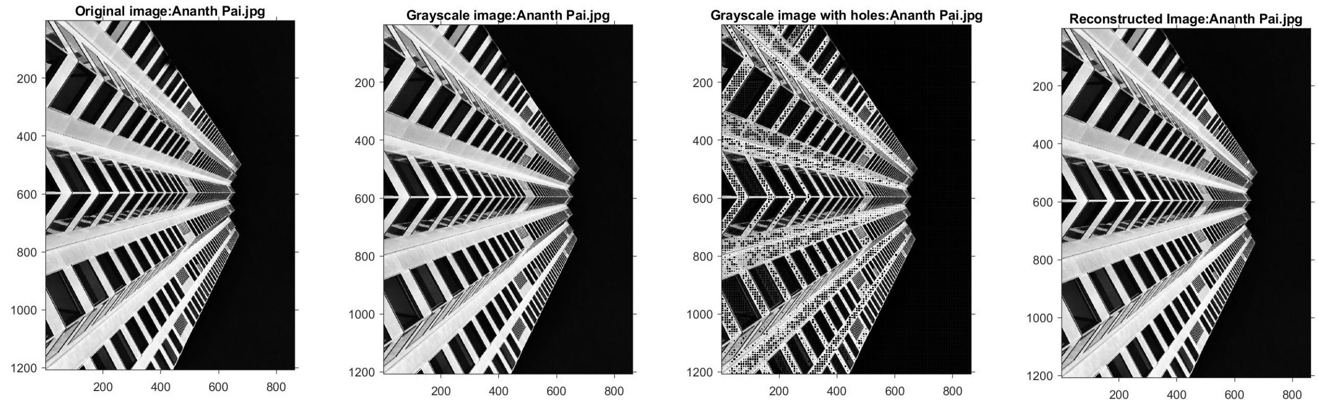


Figure 63: Ananth Pai high contrast image with 0% error introduction

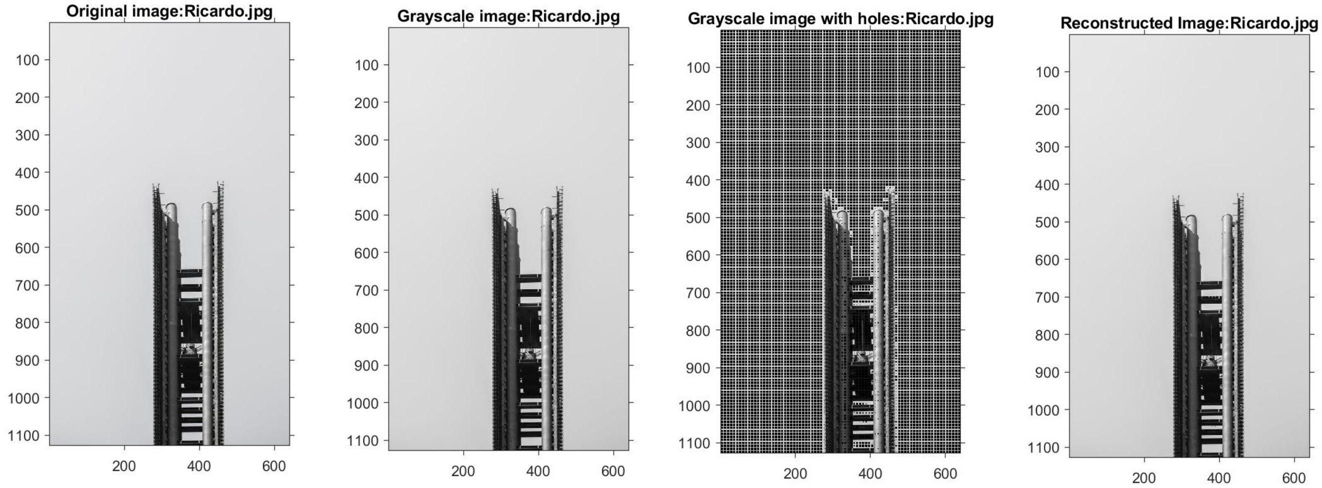


Figure 64: Ricardo high contrast image with 0% error introduction

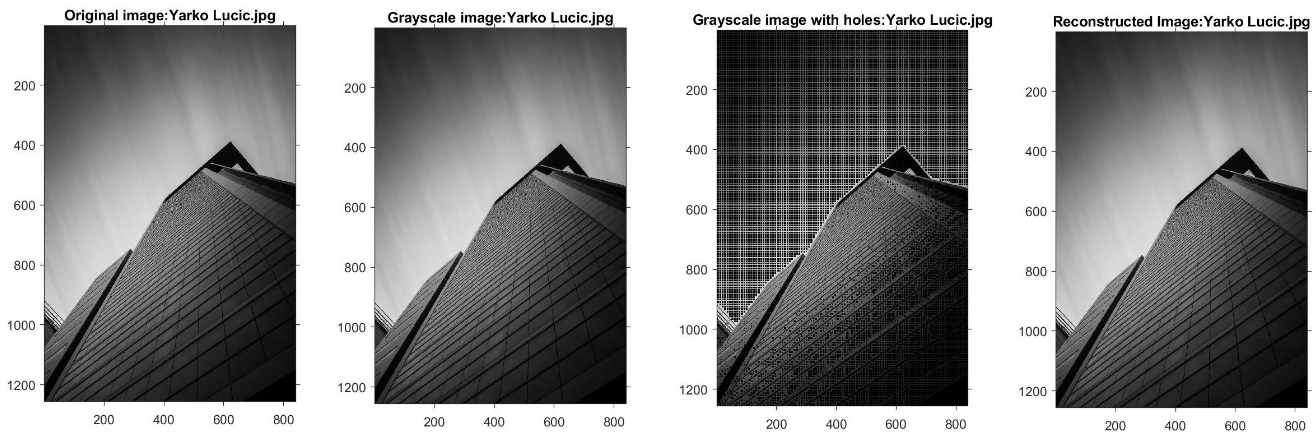


Figure 65: Yarko Lucic high contrast image with 0% error introduction

D3 Test Case 3: PSNR and MSE Values of the Different Image Types

D3.1 Pattern Image

Table 5: Table showing the PSNR and MSE for the chosen Pattern Image at increasing Error Probability with the Bit Flip Error Introduced

Error Probability (%)	**PSNR (dB)**	**MSE**
Holes Only		
0	46.090331	1.59973
20	31.500574	46.02809
40	28.524530	91.33293
60	26.826794	135.02090
80	25.534215	181.82720
100	24.513786	229.98620
DCT Only		
0	58.936025	0.08308
20	30.662334	55.82734
40	27.908455	105.25310
60	26.095628	159.77820
80	24.759635	217.32860
100	23.698977	277.44870
Holes and DCT		
0	45.058124	2.02893
20	31.474314	46.30725
40	28.554583	90.70308
60	26.756813	137.21420
80	25.505180	183.04690
100	24.609905	224.95200

Table 6: Table showing the PSNR and MSE for the chosen Pattern Image at increasing Error Probability with the Ones Compliment Error Introduced

Error Probability (%)	**PSNR (dB)**	**MSE**
Holes Only		
0	46.090331	1.59973
20	24.435847	234.15080
40	21.382830	472.93050
60	19.619495	709.79160
80	18.273401	967.69890
100	17.306828	1208.92000
DCT Only		
0	58.936025	0.08308
20	24.052951	255.73210
40	21.038030	512.00870
60	19.320175	760.43630
80	18.024578	1024.76100
100	17.062288	1278.94400
Holes and DCT		
0	45.058124	2.02893
20	24.485815	231.47220
40	21.497122	460.64690
60	19.712444	694.76190
80	18.397289	940.48420
100	17.444356	1171.23700

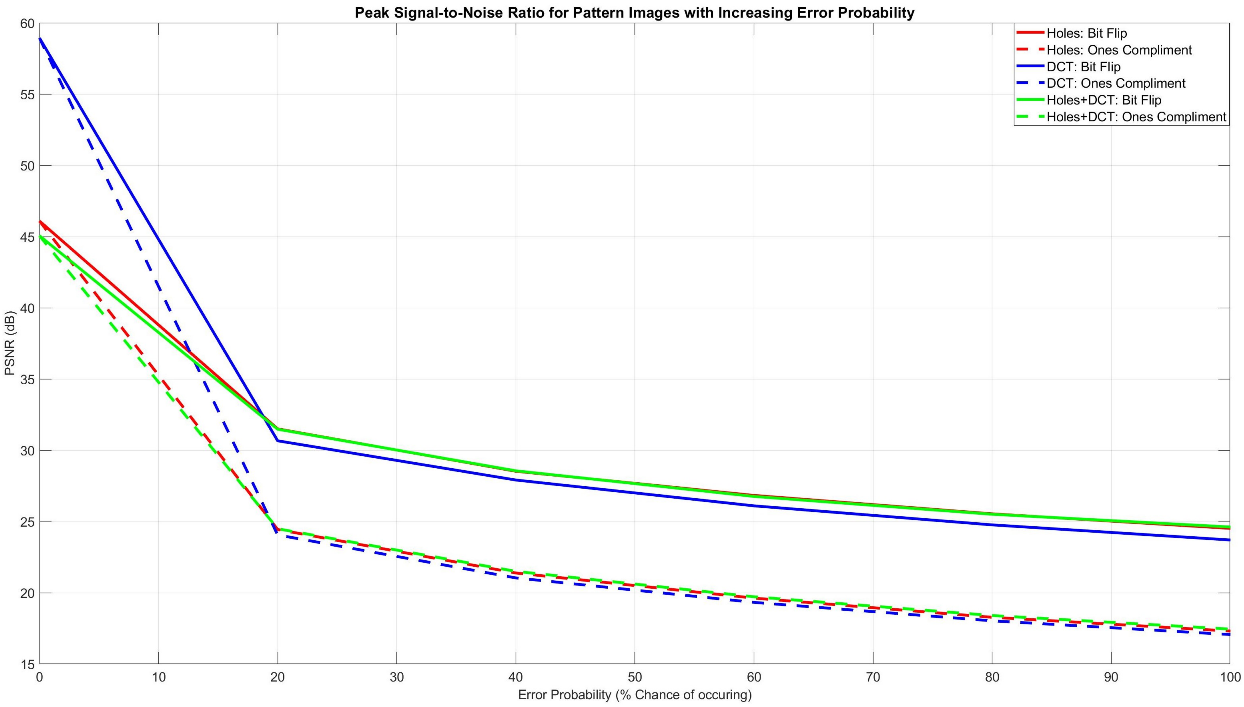


Figure 66: PSNR for the Pattern Image at increasing Error Probability

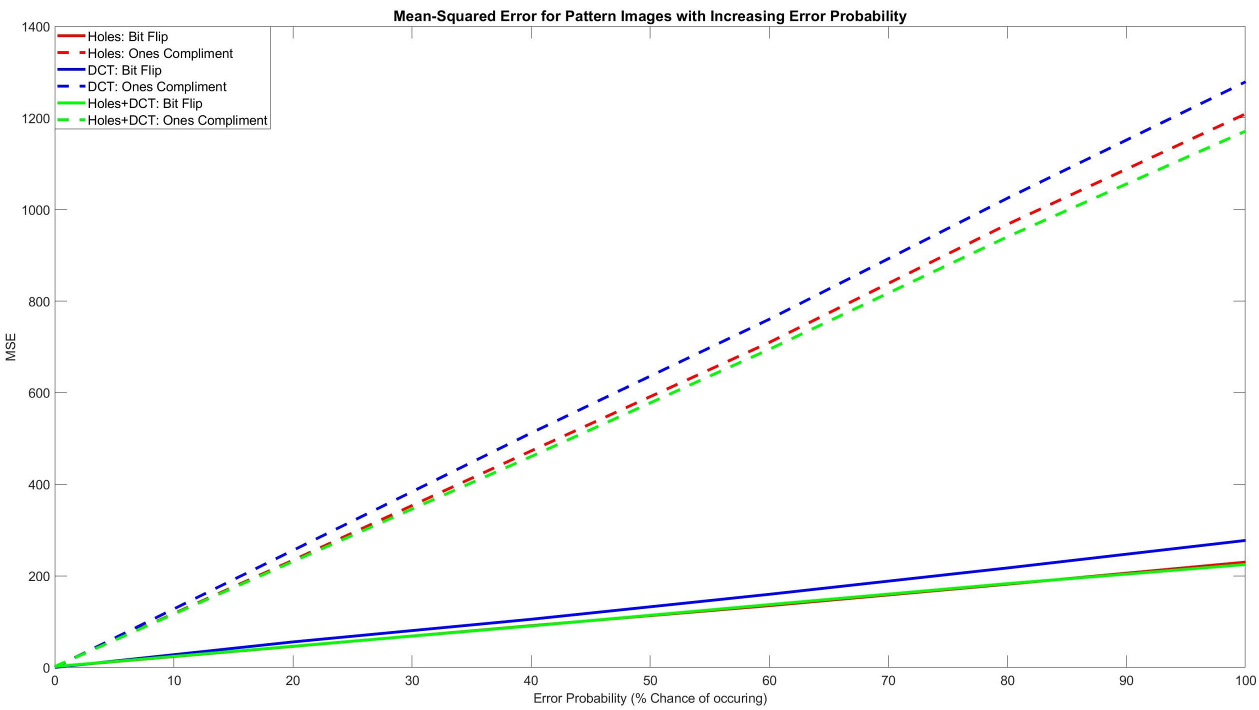


Figure 67: MSE for the Pattern Image at increasing Error Probability

D3.2 Landscape Image

Table 7: Table showing the PSNR and MSE for the chosen Landscape Image at increasing Error Probability with the Bit Flip Error Introduced

Error Probability (%)	**PSNR (dB)**	**MSE**
Holes Only		
0	44.860377	2.12345
20	32.088255	40.20267
40	29.102520	79.95183
60	27.407911	118.11070
80	26.222169	155.18990
100	25.182748	197.15400
DCT Only		
0	42.077255	4.03046
20	30.590382	56.75997
40	27.982861	103.46520
60	26.367792	150.07250
80	25.203541	196.21230
100	24.193385	247.59500
Holes and DCT		
0	40.268708	6.11237
20	32.368187	37.69308
40	29.033795	81.22708
60	27.282693	121.56570
80	26.247029	154.30410
100	25.060405	202.78690

Table 8: Table showing the PSNR and MSE for the chosen Landscape Image at increasing Error Probability with the Ones Compliment Error Introduced

Error Probability (%)	**PSNR (dB)**	**MSE**
Holes Only		
0	44.860383	2.12345
20	24.880874	211.34550
40	21.805498	429.07240
60	20.131662	630.83260
80	18.678414	881.53380
100	17.664884	1113.24800
DCT Only		
0	42.077255	4.03046
20	23.652918	280.40680
40	20.611044	564.90570
60	19.086676	802.44040
80	17.702773	1103.57800
100	16.708273	1387.56400
Holes and DCT		
0	40.268708	6.11237
20	25.084847	201.64880
40	22.023406	408.07480
60	19.889896	666.94610
80	18.782955	860.56730
100	17.753161	1090.84800

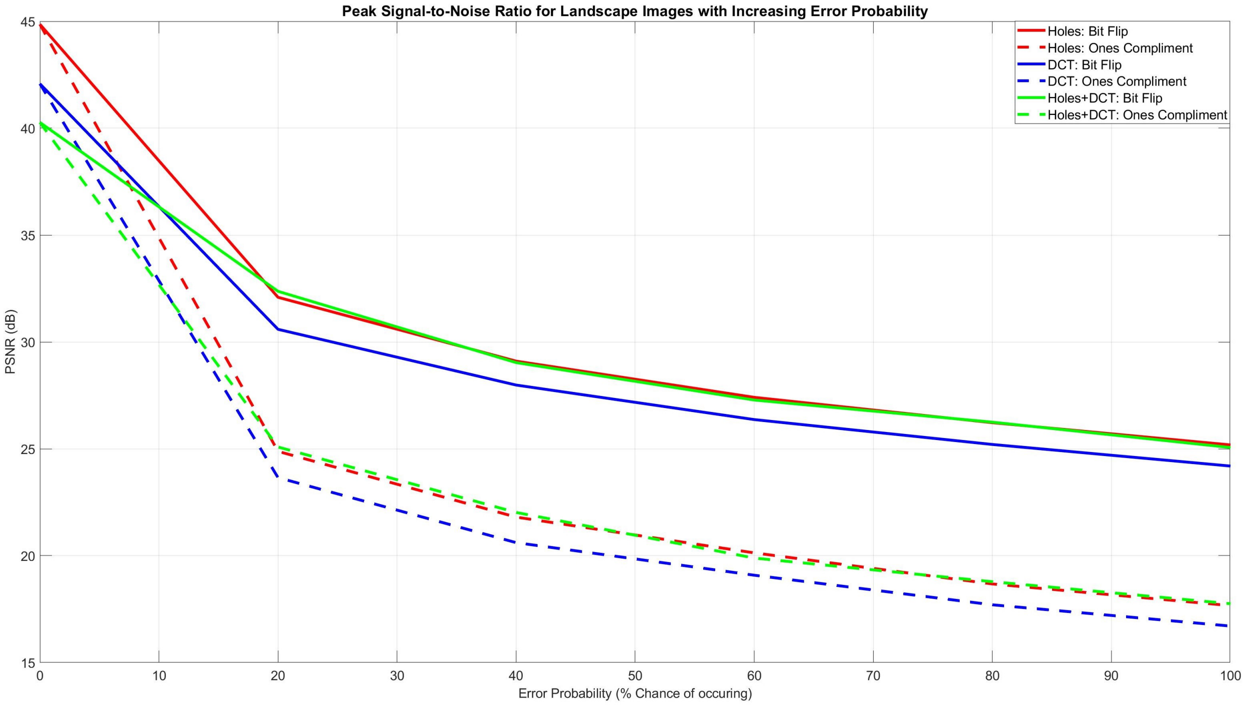


Figure 68: PSNR for the Landscape Image at increasing Error Probability

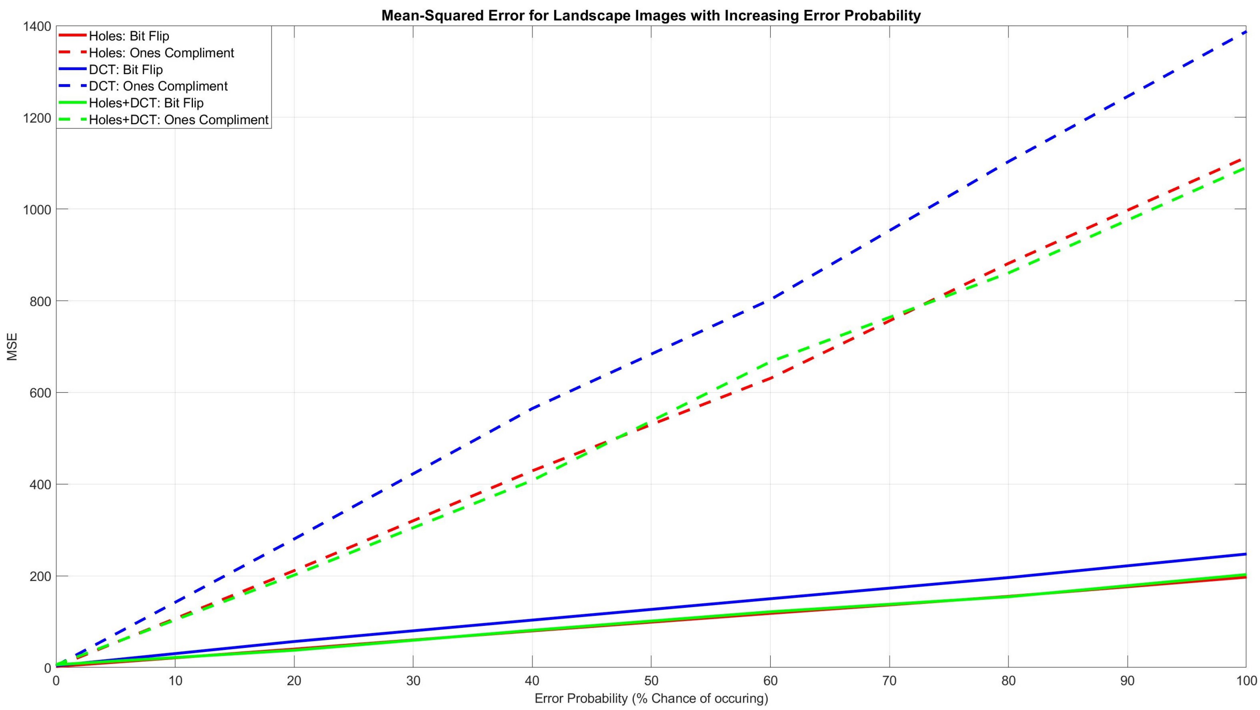


Figure 69: MSE for the Landscape Image at increasing Error Probability

D3.3 High Contrast Image

Table 9: Table showing the PSNR and MSE for the chosen High Contrast Image at increasing Error Probability with the Bit Flip Error Introduced

Error Probability (%)	**PSNR (dB)**	**MSE**
Holes Only		
0	45.250926	1.94083
20	30.423042	58.98969
40	27.896818	105.53550
60	26.129692	158.52990
80	24.759939	217.31340
100	23.796799	271.26920
DCT Only		
0	39.349911	7.55247
20	28.941095	82.97951
40	26.004712	163.15830
60	24.291938	242.03970
80	23.174823	313.03870
100	22.092474	401.63630
Holes and DCT		
0	37.952647	10.41875
20	30.223452	61.76396
40	27.330888	120.22410
60	26.063227	160.97470
80	24.438803	233.99150
100	23.688828	278.09780

Table 10: Table showing the PSNR and MSE for the chosen High Contrast Image at increasing Error Probability with the Ones Compliment Error Introduced

Error Probability (%)	**PSNR (dB)**	**MSE**
Holes Only		
0	45.250926	1.94083
20	24.552474	227.94650
40	21.162676	497.52250
60	19.412237	744.48620
80	17.912736	1051.49400
100	16.905381	1325.99600
DCT Only		
0	39.349911	7.55247
20	22.576378	359.28790
40	19.583178	715.75200
60	17.775618	1085.22200
80	16.574192	1431.07100
100	15.629156	1778.95400
Holes and DCT		
0	37.952647	10.41875
20	23.954011	261.62500
40	21.309092	481.02890
60	19.204572	780.94990
80	17.967303	1038.36500
100	16.902022	1327.02200

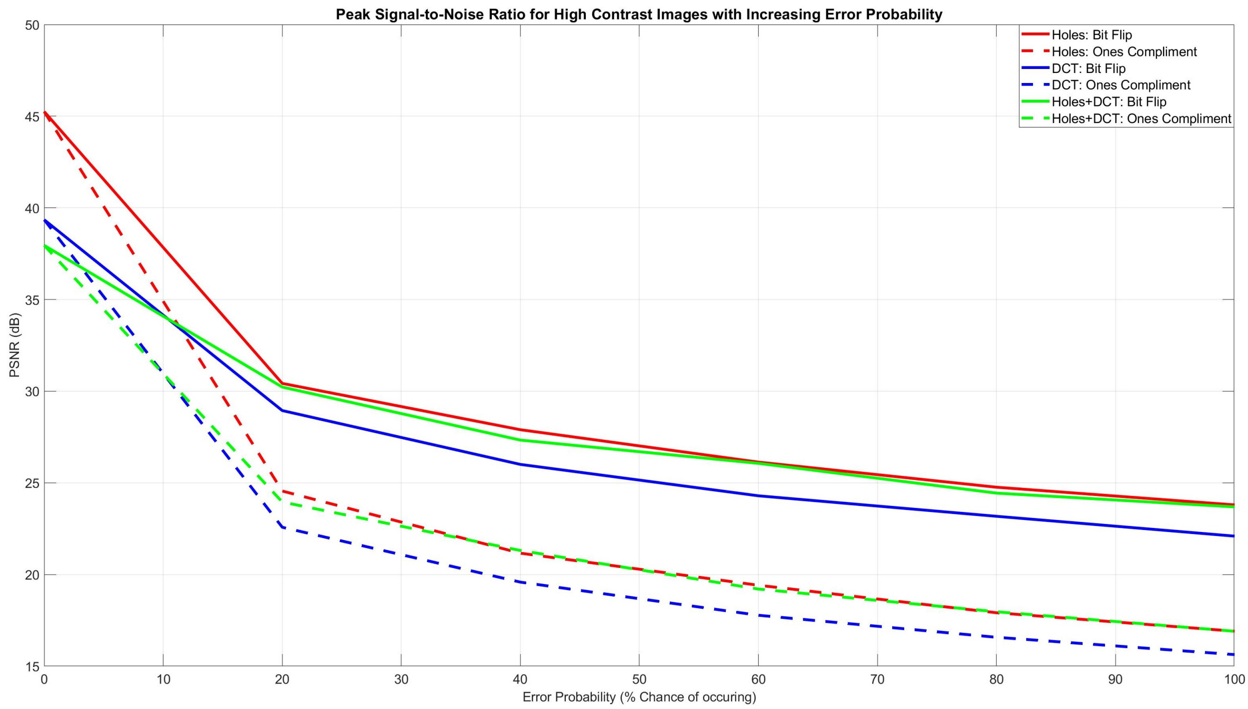


Figure 70: PSNR for the High Contrast Image at increasing Error Probability

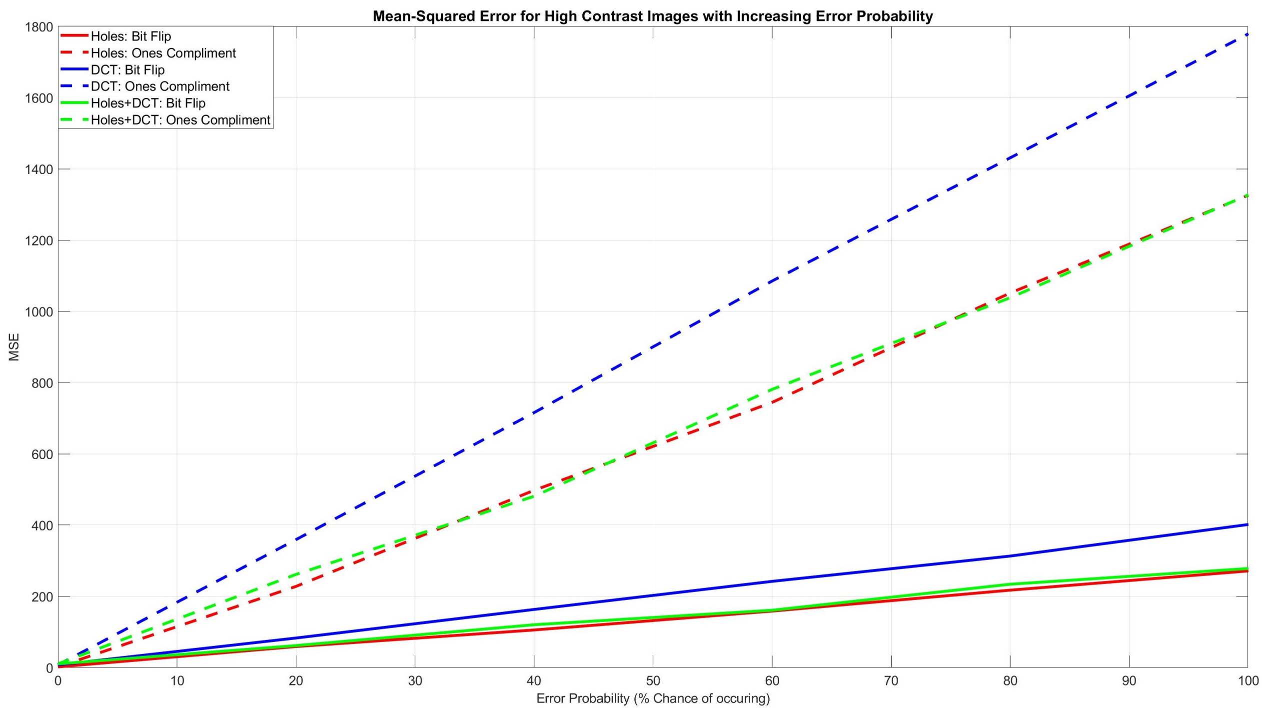


Figure 71: MSE for the High Contrast Image at increasing Error Probability

Table 11: Table showing the compression ratio, PSNR and MSE for known images using different compression schemes with 0% error probability

Holes Only			
Image	**Compression Ratio**	**PSNR**	**MSE**
Mandrill	0.9807	48.393405	0.941326
Peppers	1.0984	36.108973	15.928800
Cameraman	1.3798	38.533287	9.114895
DCT Only			
Mandrill	0.9827	25.662864	176.52
Peppers	1.0933	25.770724	172.19
Cameraman	1.2116	25.616368	178.42
DWT			
Mandrill	2.52497	35.0131	20.5006
Peppers	2.73244	34.854	21.2657
Cameraman	9.03488	35.1201	20.002

Table 12: Table showing the compression ratio, PSNR and MSE for known images using different compression schemes with 10% error probability using bit flip error introduction

Holes Only			
Image	**Compression Ratio**	**PSNR**	**MSE**
Mandrill	0.9811	34.407569	23.568040
Peppers	1.0989	32.436188	37.107490
Cameraman	1.3807	32.760065	34.440850
DCT Only			
Mandrill	0.9827	53.868945	0.2668
Peppers	1.0933	53.033403	0.3234
Cameraman	1.2116	52.770272	0.3436
DWT			
Mandrill	2.55287	25.2522	194.0265
Peppers	2.59984	24.9986	205.6916
Cameraman	9.03473	25.4041	187.357

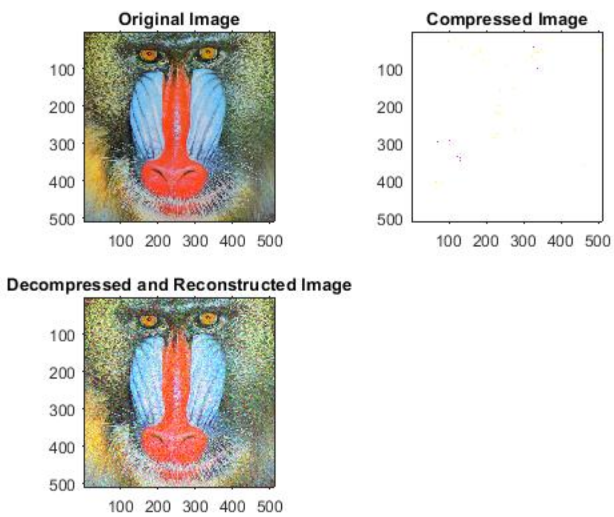


Figure 72: DWT Compression on the Mandrill image with 10% error introduction with bit flip error introduction

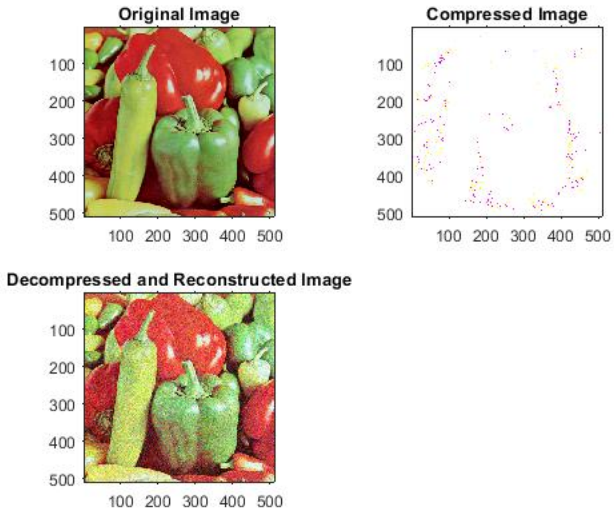


Figure 73: DWT Compression on the Peppers image with 10% error introduction with bit flip error introduction

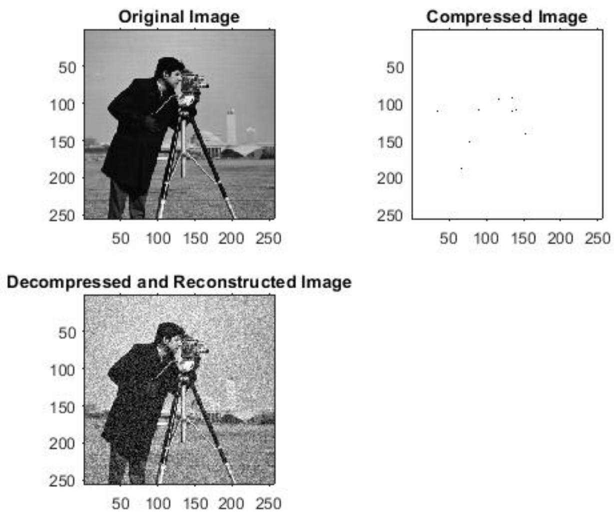


Figure 74: DWT Compression on the Cameraman image with 10% error introduction with bit flip error introduction

E SUPPORTING IMAGES

The Appendix presented here contain images that are required for detailed explanation in the report. The initial figures Figures 75 and 76 present a larger version of the system block diagram and a visual representation on how the blocks in the domain pool are indexed, respectively. Section E1 presents all the simulated images with the different Chebyshev distance values. Section E2 presents the various pattern, landscape and high contrast images that were used in the testing of this project.

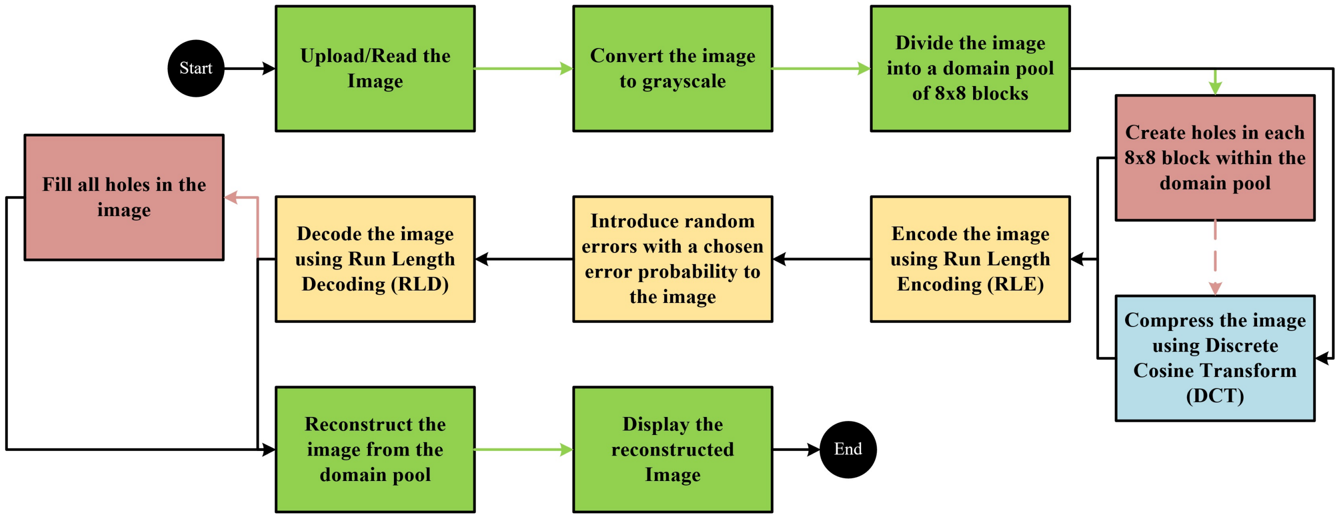


Figure 75: Block diagram of the implemented system

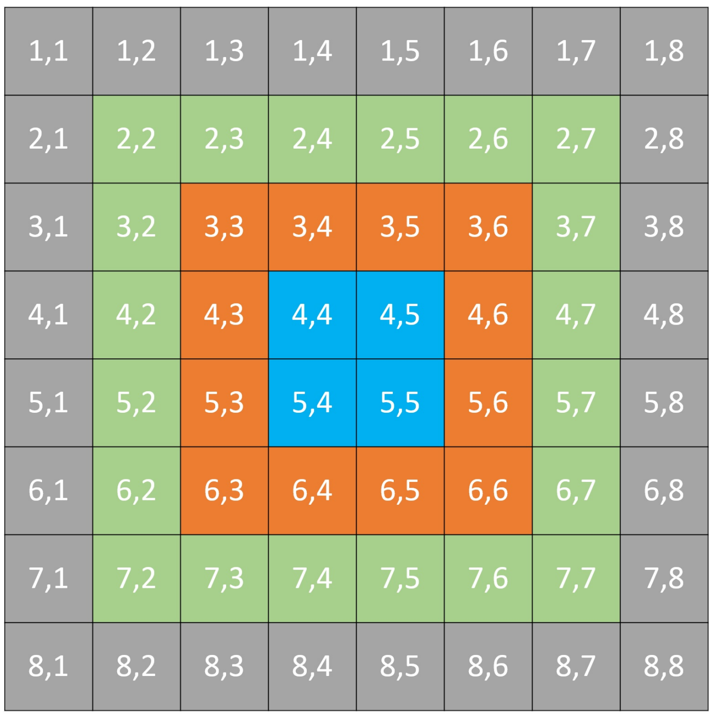


Figure 76: Indexing of each 8×8 pixel block in the domain pool

Figure 77: *Holes*-only algorithm with a Chebyshev distance check value of 1

Figure 78: *Holes*-only algorithm with a Chebyshev distance check value of 2

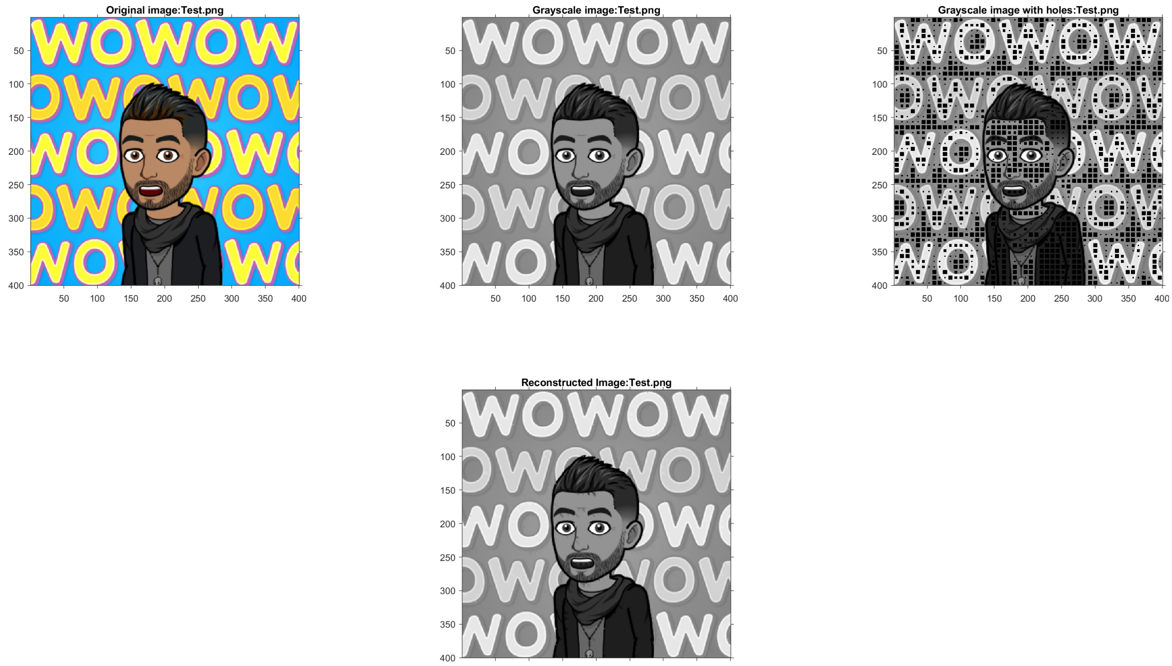


Figure 79: *Holes*-only algorithm with a Chebyshev distance check value of 3

Figure 80: *Holes*-only algorithm with a Chebyshev distance check value of 4

Figure 81: *Holes*-only algorithm with a Chebyshev distance check value of 5

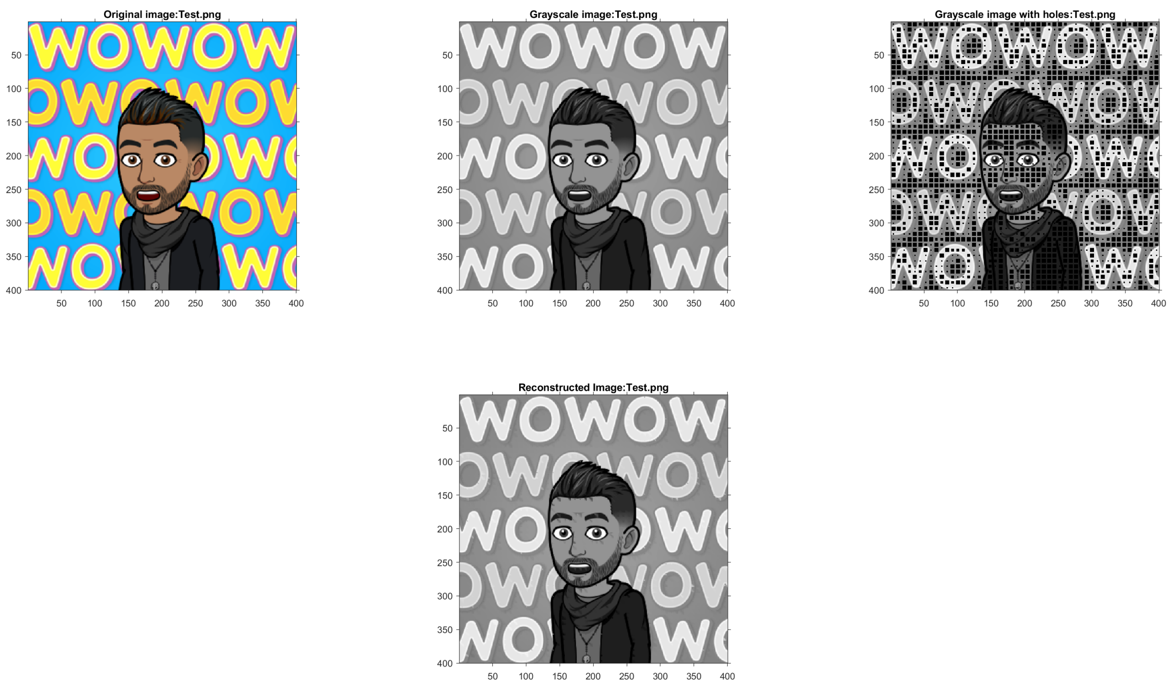


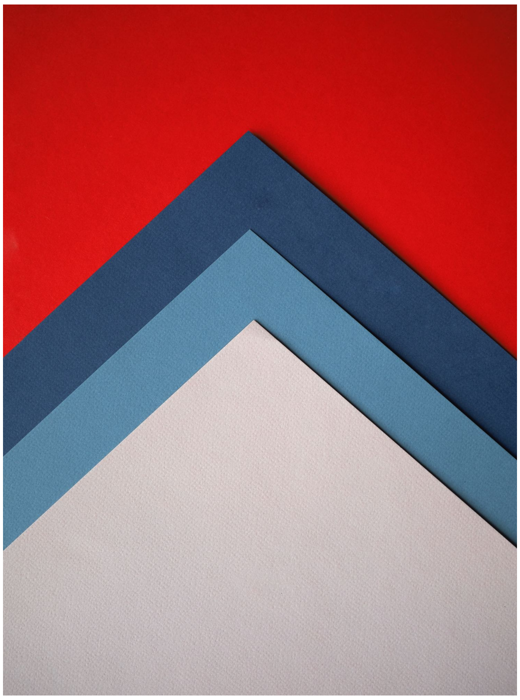
Figure 82: *Holes*-only algorithm with a Chebyshev distance check value of 6

Figure 83: *Holes*-only algorithm with a Chebyshev distance check value of 7

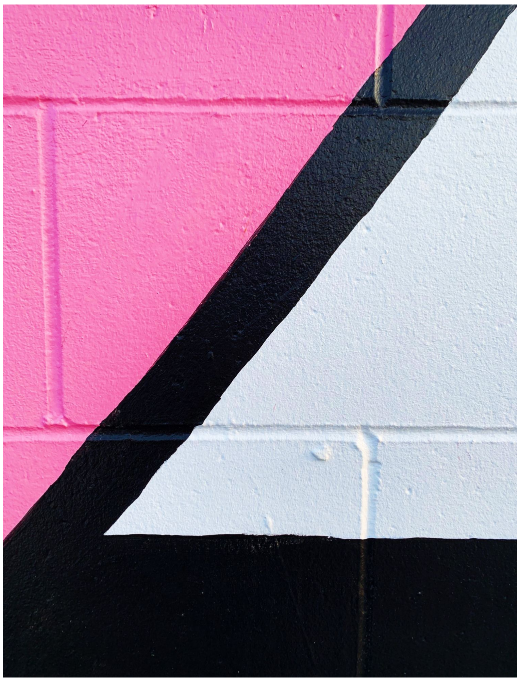
Figure 84: *Holes*-only algorithm with a Chebyshev distance check value of 8

Figure 85: *Holes*-only algorithm with a Chebyshev distance check value of 9

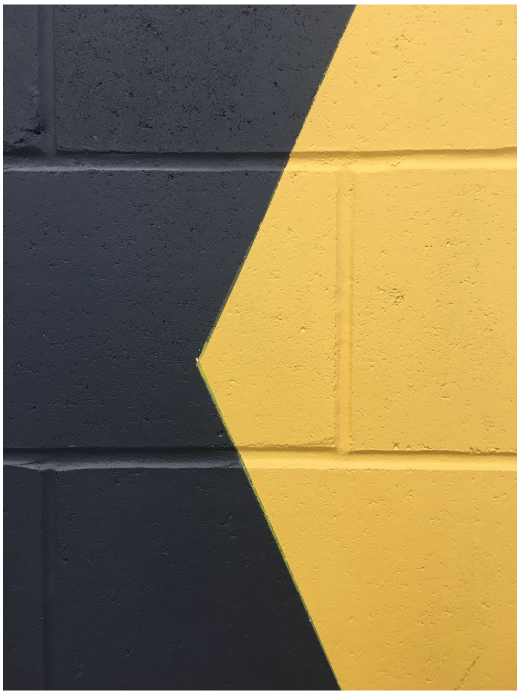
Figure 86: *Holes*-only algorithm with a Chebyshev distance check value of 10



(a)



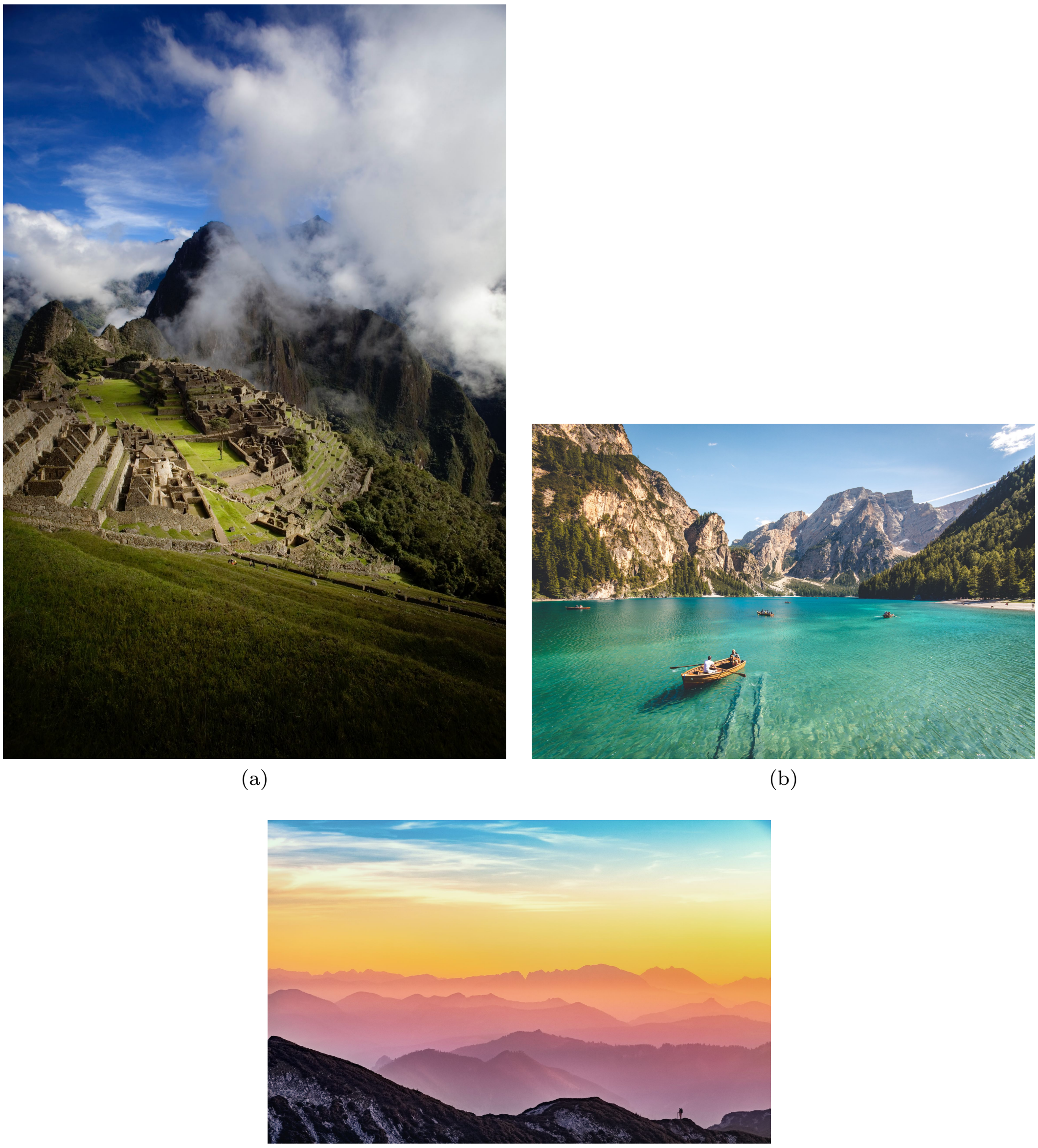
(b)



(c)

(d)

Figure 87: Pattern Images [(a) Andrej Lisakov, (b) Bryan Garces, (c) Ross Elder, (d) Vino Li]



(a)

(b)

(c)

Figure 88: Landscape Images [(a) Amanda Kerr, (b) Pietro De Grandi, (c) Simon Matzinger]

(a) (b) (c)

Figure 89: High Contrast Images [(a) Ananth Pai, (b) Ricardo, (c) Yerko Lucic]

F FORMULAE, ALGORITHMS AND LICENSES

This Appendix presents all the relevant formulae, algorithms and code that was used throughout this project implementation. Section F1 contain all relevant formulas referenced in the report, while Section F3 presents the high-level algorithms for the creation and filling of holes. The final section, Section F2 presents the license for the use of the adapted Run Length Encoding and Decoding.

F1 Formulae

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \tag{1}$$

$$coordinateValue = 8(blockIndex) - 7 \tag{2}$$

$$D_{Chebyshev}(p,q) := \max_i(|p_i - q_i|) \tag{3}$$

$$MSE = \frac{1}{ab} \sum_{i=0}^{a-1} \sum_{j=0}^{b-1} [X(i,j) - Y(i,j)]^2 \tag{4}$$

$$PSNR = 20 \cdot log_{10}(255) - 10 \cdot log_{10}(MSE) \tag{5}$$

$$CR = \frac{No.\ of\ bits\ in\ uncompressed\ image}{No.\ of\ bits\ transmitted\ after\ encoding} \tag{6}$$

F2 License

Listing 1: Copyright License for use of the code

F3 Algorithms

Algorithm 1 High-level algorithm to create holes in a `8x8` blocks

```
for every block in the domain pool do
    Set n=2
    Go to nxn square in 8x8 block
    Calculate average of pixels in nxn
    for every pixel in the nxn block do
        Check Chebyshev distance between pixels (y[]) and average value (x)
    end for
    if Chebyshev distance between x and each value of y[] < 6 then
        Repeat For n=4
        if Chebyshev distance between x and each value of y[] < 6 then
            Repeat For n=6
            if Chebyshev distance between x and each value of y[] < 6 then
                Create hole in 6x6
            else
                Create hole in 4x4
            end if
        else
            Create hole in 2x2
        end if
    else
        Move to next block in domain pool
    end if
end for
```

Algorithm 2 High-level algorithm for filling holes in `8x8` blocks

```
Set n=2
Go to nxn square in 8x8 block
Calculate average of pixels in nxn
for every pixel in the nxn block do
    Check Chebyshev distance between pixels (y[]) and average value (x)
end for
if Chebyshev distance between x and each value of y[] < 6 then
    Repeat For n=4
    if Chebyshev distance between x and each value of y[] < 6 then
        Repeat For n=6
        if Chebyshev distance between x and each value of y[] < 6 then
            Move to top left of nxn block
            Calculate average of pixels directly:above, left & above-left. Fill pixel with average value
            Proceed from left to right and top to bottom
        else
            Fill hole in 4x4
        end if
    else
        Fill hole in 2x2
    end if
else
    Move to next block in domain pool
end if
```

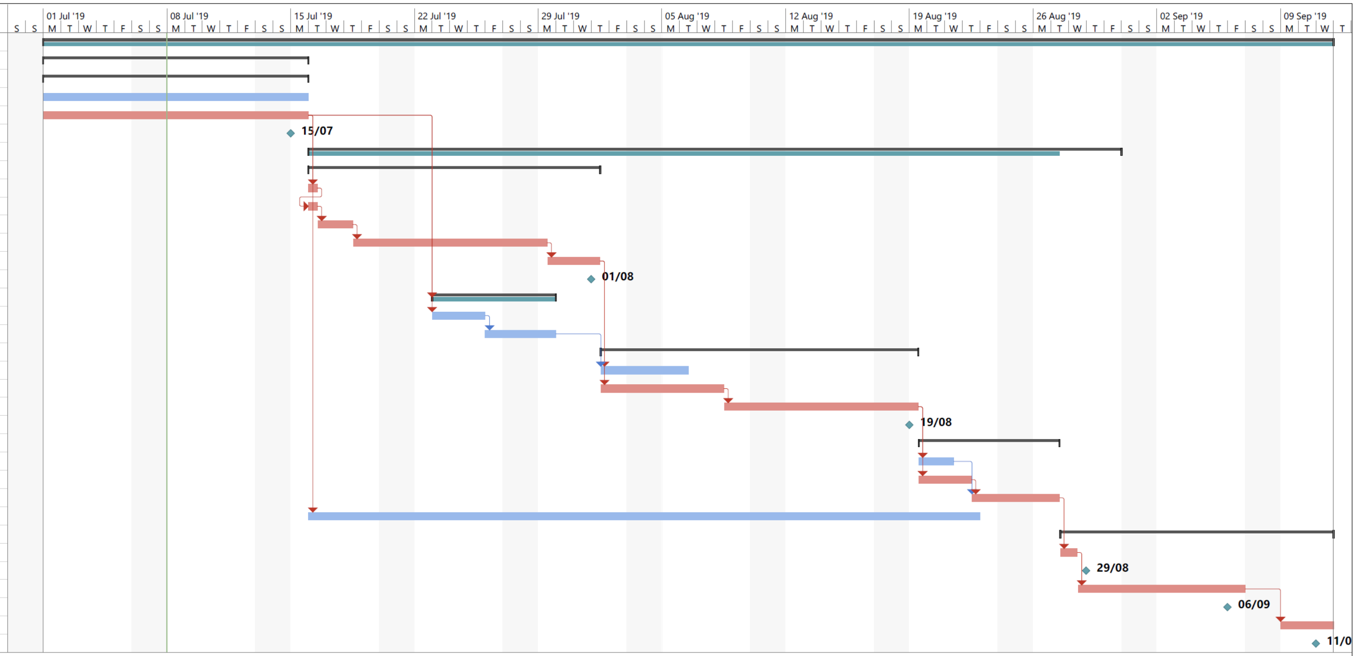
G PROJECT MANAGEMENT

This Appendix contains all material relating to Project Management. The first part shows the Table of tasks and the Gantt chart that was created in the Project Planning phase. Following this, Section G1 shows the individually recorded minutes for this project written by the author and his colleague. Section G2 shows the minutes for the entire Telecommunication's Group Meetings which were held weekly and recorded by different groups each week.

	Task Name	Duration	Start	Finish	Predecessors
1	**Lab Project: Image Compression**	**53 days**	**Mon 01/07/19**	**Wed 11/09/19**	
2	**Planning**	**11 days**	**Mon 01/07/19**	**Mon 15/07/19**	
3	**Research**	**11 days**	**Mon 01/07/19**	**Mon 15/07/19**	
4	Literature Review	11 days	Mon 01/07/19	Mon 15/07/19	
5	Project Plan	11 days	Mon 01/07/19	Mon 15/07/19	
6	Project Plan submission	0 days	Mon 15/07/19	Mon 15/07/19	
7	**Project Execution**	**34 days**	**Tue 16/07/19**	**Fri 30/08/19**	
8	**Encoder Implementation**	**12.5 days**	**Tue 16/07/19**	**Thu 01/08/19**	
9	Upload Image	0.5 days	Tue 16/07/19	Tue 16/07/19	5
10	Convert Image to grayscale	0.5 days	Tue 16/07/19	Tue 16/07/19	9
11	Divide image into domain pool	2 days	Tue 16/07/19	Thu 18/07/19	10
12	Develop hole-creation algorithm	7 days	Thu 18/07/19	Mon 29/07/19	11
13	Compress image using DCT	3 days	Mon 29/07/19	Thu 01/08/19	12
14	Complete Encoder Side	0 days	Thu 01/08/19	Thu 01/08/19	
15	**Channel Implementation**	**5 days**	**Tue 23/07/19**	**Mon 29/07/19**	**5**
16	Set up channel	3 days	Tue 23/07/19	Thu 25/07/19	5
17	Create error probability	2 days	Fri 26/07/19	Mon 29/07/19	16
18	**Decoder Implementation**	**12 days**	**Thu 01/08/19**	**Mon 19/08/19**	
19	Decode and introduce errors	3 days	Thu 01/08/19	Tue 06/08/19	13,17
20	Decompress using inverse DCT	5 days	Thu 01/08/19	Thu 08/08/19	13
21	Fill holes in image	7 days	Thu 08/08/19	Mon 19/08/19	20
22	Complete Decoder side	0 days	Mon 19/08/19	Mon 19/08/19	
23	**Performance Index**	**6 days**	**Mon 19/08/19**	**Tue 27/08/19**	
24	Generate results for DCT compression	2 days	Mon 19/08/19	Wed 21/08/19	21
25	Implement second compression technique	3 days	Mon 19/08/19	Thu 22/08/19	21
26	Final results comparison	3 days	Thu 22/08/19	Tue 27/08/19	24,25
27	Monitoring and Control	28 days	Tue 16/07/19	Thu 22/08/19	5
28	**Project Close**	**11.5 days**	**Tue 27/08/19**	**Wed 11/09/19**	
29	Open Day (Build Demonstration)	1 day	Tue 27/08/19	Wed 28/08/19	26
30	Open Day	0 days	Thu 29/08/19	Thu 29/08/19	
31	Final report	7.5 days	Wed 28/08/19	Fri 06/09/19	29
32	Final report submission	0 days	Fri 06/09/19	Fri 06/09/19	
33	Final presentation	3 days	Mon 09/09/19	Wed 11/09/19	31
34	Final presentation submission	0 days	Wed 11/09/19	Wed 11/09/19	

Figure 90: Table showing the tasks and time allocated in completing the project

ID	Task Name	Duration	Start	Finish
1	**Lab Project: Image Compression**	**53 days**	**Mon 01/07/19**	**Wed 11/09/19**
2	**Planning**	**11 days**	**Mon 01/07/19**	**Mon 15/07/19**
3	**Research**	**11 days**	**Mon 01/07/19**	**Mon 15/07/19**
4	Literature Review	11 days	Mon 01/07/19	Mon 15/07/19
5	Project Plan	11 days	Mon 01/07/19	Mon 15/07/19
6	Project Plan submission	0 days	Mon 15/07/19	Mon 15/07/19
7	**Project Execution**	**34 days**	**Tue 16/07/19**	**Fri 30/08/19**
8	**Encoder Implementation**	**12.5 days**	**Tue 16/07/19**	**Thu 01/08/19**
9	Upload Image	0.5 days	Tue 16/07/19	Tue 16/07/19
10	Convert Image to grayscale	0.5 days	Tue 16/07/19	Tue 16/07/19
11	Divide image into domain pool	2 days	Tue 16/07/19	Thu 18/07/19
12	Develop hole-creation algorithm	7 days	Thu 18/07/19	Mon 29/07/19
13	Compress image using DCT	3 days	Mon 29/07/19	Thu 01/08/19
14	Complete Encoder Side	0 days	Thu 01/08/19	Thu 01/08/19
15	**Channel Implementation**	**5 days**	**Tue 23/07/19**	**Mon 29/07/19**
16	Set up channel	3 days	Tue 23/07/19	Thu 25/07/19
17	Create error probability	2 days	Fri 26/07/19	Mon 29/07/19
18	**Decoder Implementation**	**12 days**	**Thu 01/08/19**	**Mon 19/08/19**
19	Decode and introduce errors	3 days	Thu 01/08/19	Tue 06/08/19
20	Decompress using inverse DCT	5 days	Thu 01/08/19	Thu 08/08/19
21	Fill holes in image	7 days	Thu 08/08/19	Mon 19/08/19
22	Complete Decoder side	0 days	Mon 19/08/19	Mon 19/08/19
23	**Performance Index**	**6 days**	**Mon 19/08/19**	**Tue 27/08/19**
24	Generate results for DCT compression	2 days	Mon 19/08/19	Wed 21/08/19
25	Implement second compression techniqu	3 days	Mon 19/08/19	Thu 22/08/19
26	Final results comparison	3 days	Thu 22/08/19	Tue 27/08/19
27	Monitoring and Control	28 days	Tue 16/07/19	Thu 22/08/19
28	**Project Close**	**11.5 days**	**Tue 27/08/19**	**Wed 11/09/19**
29	Open Day (Build Demonstration)	1 day	Tue 27/08/19	Wed 28/08/19
30	Open Day	0 days	Thu 29/08/19	Thu 29/08/19
31	Final report	7.5 days	Wed 28/08/19	Fri 06/09/19
32	Final report submission	0 days	Fri 06/09/19	Fri 06/09/19
33	Final presentation	3 days	Mon 09/09/19	Wed 11/09/19
34	Final presentation submission	0 days	Wed 11/09/19	Wed 11/09/19



Project: Schedule
Date: Mon 08/07/19

Task | Split | Milestone | Summary | Project Summary | Inactive Task | Inactive Milestone | Inactive Summary | Manual Task | Duration-only | Manual Summary Rollup | Manual Summary | Start-only | Finish-only | External Tasks | External Milestone | Deadline | Critical | Critical Split | Progress | Manual Progress

Figure 91: Gantt chart showing the tasks and time allocated in completing the project with the critical path highlighted

G1 Individually Recorded Meeting Minutes

The individually recorded minutes can be found from the next page onwards.

INITIAL MEETING FOR PROJECT PLAN

Location:	University of the Witwatersrand, Chamber of Mines
Date:	28 June 2019
Time:	11:30
Facilitator:	Prof. F. Takawira

Agenda Items & Discussion Points

1. Prof. Takawria will be in China in the coming weeks and will not be available for consultation for the first submission of the Project Plan.
2. The Project Plan needs to be created for the upcoming weeks so that Prof can decide with us whether our approach is correct.
3. Without Prof being available, we will need to start the project and begin coding the project with the least amount of assistance.
4. Main focus is to create the Project Plan.
5. The Project Plan needs to cover the entire period of the project.
6. As much detail as possible is required in the Plan itself. Where the algorithm must be defined and explained, but no pseudocode needs to be given.

Action Items	Owner(s)	Deadline	Status
Describe the steps taken in the project.	Kishan Narotam & Nitesh Nana	01 July 2019	Needs to be started

PROJECT PLAN MEETING

Location:	University of the Witwatersrand, Chamber of Mines
Date:	01 July 2019
Time:	08:45
Facilitator:	Prof. F Takawira

Agenda Items & Discussion Points

1. The basic Project Plan that was created is not sufficient as not enough detail is given,
2. Prof has given a clear clarification of what is required in the project, i.e. the image will undergo 2 compression techniques.
3. The actual compression must be done using known techniques such as DCT, etc.
4. The creation of holes was a main focus point of the discussion.
5. The breakdown of the encoding was discussed and a possible way forward was suggested (DCT then holes being created)
6. How the image will be transmitted was discussed and how to introduce errors.
7. The overall idea is decent, but a more step-by-step explanation is needed.
8. Breakdown of approach must be done as soon as possible, so that a decision can be made on which steps can be completed while Prof is away.

Action Items	Owner(s)	Deadline	Status
Describe the steps taken in the project	Kishan Narotam & Nitesh Nana	01 July 2019	Completed
Complete Project Plan	Kishan Narotam & Nitesh Nana	15 July 2019	In Progress
Breakdown the implementation into steps	Kishan Narotam & Nitesh Nana	02 July 2019	Started and In Progress
Research other image compression techniques	Kishan Narotam & Nitesh Nana	15 July 2019	In Progress

FINAL PROJECT PLAN

Location:	University of the Witwatersrand, Chamber of Mines
Date:	04 July 2019
Time:	09:30
Facilitator:	Prof. F Takawira

Agenda Items & Discussion Points

1. A more detailed and descriptive project plan was presented to Prof.
2. The overall idea and plan were given provisional acceptance.
3. The idea behind creating the holes in an image may be too simple, so a new suggestion was given: to research other *similarity indexes*.
4. The compression technique of DCT can be implemented with relative ease in MATLAB, thus in order to make the project more "meatier", we will be required to add another method of compressing as a way of comparing the results.
5. The second method of compression chosen will be Fractal compression, as it can be implemented with relative ease in MATLAB.
6. The project can be started while Prof is currently not in the country with clear way forward.

Action Items	Owner(s)	Deadline	Status
Complete Project Plan (for final submission)	Kishan Narotam & Nitesh Nana	15 July 2019	In Progress
Breakdown the implementation into steps	Kishan Narotam & Nitesh Nana	02 July 2019	Completed
Research other image compression techniques	Kishan Narotam & Nitesh Nana	15 July 2019	Completed
Research other types of similarity indices	Kishan Narotam & Nitesh Nana	ASAP	Started and In Progress

Begin implementation of the project	Kishan Narotam & Nitesh Nana	ASAP	Needs to be started
Complete as much of the DCT implementation	Kishan Narotam & Nitesh Nana	Month-end	Needs to be started

TELECOMMUNICATION GROUP MEETING 1

Location: University of the Witwatersrand, Chamber of Mines, Convergence Lab

Date: 16 July 2019

Time: 14:00

Facilitator: Group 19G14
Prof. Ling Cheng

Agenda Items & Discussion Points

1. All groups must introduce themselves, supervisors, and what their respective projects are.
2. How the meetings will be run and who will be keeping the minutes each week.
3. A different group will chair the meeting each week and will keep minutes that will be accessible to everyone through a shared Google document.
4. The projects are discussed and the basics of the approaches and problems are mentioned with a few plausible solutions given.
5. The groups within the larger telecommunications include: 19G01, 19G08, 19G14, 19G18, 19G21, 19G23, 19G28, 19G41,

Action Items	Owner(s)	Deadline	Status
Complete Project Plan	Kishan Narotam & Nitesh Nana	15 July 2019	Completed
Begin implementation of the project	Kishan Narotam & Nitesh Nana	ASAP	Completed
Continue with the DCT implementation of the project	Kishan Narotam & Nitesh Nana	Month-end	In Progress

TELECOMMUNICATION GROUP MEETING 2

Location: University of the Witwatersrand, Chamber of Mines, Convergence Lab

Date: 23 July 2019

Time: 12:00

Facilitator: Group 19G41

Agenda Items & Discussion Points

1. Weekly meetings may be moved to a more formal venue, where Prof. F Takawira suggesting a conference room. Destination subject to change.
2. Meeting minutes will be typed up by the current chair group allocated to that given week,
3. Discussion into the biggest problems faced by the various groups during the last week with possible solutions given out by both the supervisors and groups during who have had similar issues or whose projects falls within the same field.
4. Group 19G01 were selected to chair next week's meeting.
5. Meeting adjourned at 13:13.

Action Items	Owner(s)	Deadline	Status
Continues with the DCT implementation of the project	Kishan Narotam & Nitesh Nana	Month-End	Completed
Tweak DCT implementation	Kishan Narotam & Nitesh Nana	26 July 2019	In Progress
Create implementation for holes only	Kishan Narotam & Nitesh Nana	26 July 2019	In Progress

PROJECT UPDATE MEETING

Location:	University of the Witwatersrand, Chamber of Mines
Date:	25 July 2019
Time:	11:00
Facilitator:	Prof. F Takawira

Agenda Items & Discussion Points

1. Update Prof. on the work that was done while he was away in China,
2. Discuss with him the current issues we are having and if he has any advice.
3. Ask him how the encoding and error introduction must be done.
4. The implementation of the holes and filling the holes seems to be working well. Prof. seems pleased with it.
5. Research must be done to see and discuss why Chebyshev distance was chosen as well as the value of the distance.
6. The implementation of the DCT can be kept at high quality and low compression as the actual compression of the holes algorithm will be done with encoding.
7. Run-length encoding possibilities were discussed and the way to approach it for an image is given.

Action Items	Owner(s)	Deadline	Status
Tweak DCT implementation	Kishan Narotam & Nitesh Nana	26 July 2019	Completed
Create Implementation for holes only	Kishan Narotam & Nitesh Nana	26 July 2019	Completed
Start encoding using run-length encoding	Kishan Narotam & Nitesh Nana	02 August 2019	Needs to be started
Research and find a reason for Chebyshev distance	Kishan Narotam & Nitesh Nana	26 July 2019	In Progress
Continue working on the project	Kishan Narotam & Nitesh Nana	29 August 2019	In Progress

TELECOMMUNICATION GROUP MEETING 3

Location: University of the Witwatersrand, Chamber of Mines, Convergence Lab

Date: 30 July 2019

Time: 12:00

Facilitator: Group 19G01

Agenda Items & Discussion Points

1. Updates given on the problems discussed and mentioned from last week.
2. Any new issues or problems encountered must be stated and as the previous week, discussed with the group. Possible solutions will be given and alternatives may be stated.
3. The project is halfway through completion and by this time, projects should be reaching the end stages, and groups should start getting results for the presentation and begin working on the report.
4. Group 19G18 have been nominated to chair next week's meeting.
5. The venue will not change from Convergence Lab
6. The meeting was adjourned at 12:53

Action Items	Owner(s)	Deadline	Status
Resolve DCT Implementation issue	Kishan Narotam & Nitesh Nana	Month-End	Completed
Implement Run-length encoding and decoding for the project	Kishan Narotam & Nitesh Nana	9 August 2019	Needs to be started

TELECOMMUNICATION GROUP MEETING 4

Location: University of the Witwatersrand, Chamber of Mines, Convergence Lab

Date: 06 August 2019

Time: 12:00

Facilitator: Group 19G18

Agenda Items & Discussion Points

1. Updates given on the problems discussed and mentioned from last week.
2. Any new issues or problems encountered must be stated and as the previous week, discussed with the group. Possible solutions will be given and alternatives may be stated.
3. The project is halfway through completion and by this time, projects should be reaching the end stages, and groups should start getting results for the presentation and begin working on the report.
4. Group 19G08 have been nominated to chair next week's meeting.
5. The venue will not change from Convergence Lab
6. A question was raised regarding ethics and using certain programs and the fear of plagiarism and copyright infringement. The supervisors have given their opinion regarding it, and should a license be given, it must be shown in the final report.
7. Prof. Takawira suggested that with the project reaching it's final stages, and near completion the upcoming meetings should rather focus on the groups that require the most assistance,
8. The meeting was adjourned at 12:53

Action Items	Owner(s)	Deadline	Status
Run-length encoding algorithm needs rectifying	Kishan Narotam & Nitesh Nana	9 August 2019	Completed
Decide on a way forward in terms of attaining results	Kishan Narotam & Nitesh Nana	12 August 2019	In progress

TELECOMMUNICATION GROUP MEETING 5

Location: University of the Witwatersrand, Chamber of Mines, Convergence Lab

Date: 13 August 2019

Time: 12:00

Facilitator: Group 19G08

Agenda Items & Discussion Points

1. There is a presentation being given for the Telecoms group which all students in the Telecoms project group are invited to. An RSVP to Thabiso is required.
2. The meeting format will be changed to allow more focus to be given to groups that are struggling with their projects.
3. Updates given on the problems discussed and mentioned from last week.
4. Groups should put a lot of thought into how they are going to present for the final examination.
5. The assessment uses a rubric, pay attention to the rubric because the examiners will ask questions based on the rubric.
6. There will be no meeting for the week of open day.
7. 19G13 have been nominated to chair next week's meeting.
8. The meeting was adjourned at 12:45.

Action Items	Owner(s)	Deadline	Status
Decide on a way forward in terms of attaining results	Kishan Narotam & Nitesh Nana	12 August 2019	Complete
Find images online to test the program	Kishan Narotam & Nitesh Nana	19 August 2019	In Progress
Test the program using the various images using different methods	Kishan Narotam & Nitesh Nana	21 August 2019	In Progress
Begin design on the poster for Open Day	Kishan Narotam & Nitesh Nana	26 August 2019	Needs to be started

PROJECT UPDATE MEETING

Location:	University of the Witwatersrand, Chamber of Mines Building
Date:	16 August 2019
Time:	12:00
Facilitator:	Prof. F Takawira

Agenda Items & Discussion Points

1. With the implementation of the project near completion, this meeting is to bring Prof. Takawira up to date with the implementation and the look of the results.
2. Prof. Takawira looked into the way the image was being encoded and compressed and questioned whether the way our Run-length encoding scheme was actually compressing the image or not.
3. A suggestion was given on encoding the image so that the current run-length algorithm can be changed to ensure compression is seen. The reason being is that the current algorithm sends the values and count of those respective values through the channel.
4. The fractal compression technique may cause unnecessary stress in implementing now, if we decide, we can use an already implemented method that can be found on MATLAB's File Exchange.
5. The code needs to be completed by next week's meeting so that we can discuss with Prof. Takawira what we will present on Open Day: on the poster, on the computer as a live demonstration and results for the final report.

Action Items	Owner(s)	Deadline	Status
Find images online to test the program	Kishan Narotam & Nitesh Nana	19 August 2019	In Progress
Test the program using the various images using different methods.	Kishan Narotam & Nitesh Nana	21 August 2019	In Progress
Begin design on the poster for Open Day	Kishan Narotam & Nitesh Nana	26 August 2019	Needs to be started
Test and verify the run-length encoder used	Kishan Narotam & Nitesh Nana	18 August 2019	In Progress

TELECOMMUNICATIONS GROUP MEETING 6

Location:	University of the Witwatersrand, Chamber of Mines, Convergence Lab
Date:	20 August 2019
Time:	12:00
Facilitator:	Group 19G13

Agenda Items & Discussion Points

1. Updates given on the problems discussed and mentioned from last week.
2. Lecturers gave advice about open day and the general content that should be in the poster.
3. The assessment uses a rubric, pay attention to the rubric because the examiners will ask questions based on the rubric.
4. There is a possible lecture about doing presentations that will take place on the week of presentations
5. There will be no meeting for the week of open day.

Action Items	Owner(s)	Deadline	Status
Test and verify the run-length encoder used	Kishan Narotam & Nitesh Nana	18 August 2019	In Progress
Generate results	Kishan Narotam & Nitesh Nana	ASAP	Needs to be started

FINAL PROJECT MEETING

Location:	University of the Witwatersrand, Chamber of Mines
Date:	22 August 2019
Time:	12:30
Facilitator:	Prof. F Takawira

Agenda Items & Discussion Points

1. Test cases have been presented to Prof. Takawira, and he has given us approval for it.
2. Prof. Takawira suggests focusing on the poster design and making sure we present the correct information, such as block diagrams, graphs, etc.
3. The poster will dictate what results we should generate.
4. Overall, Prof. Takawira seems pleased with the approach we have taken and test cases we have provided.

Action Items	Owner(s)	Deadline	Status
Design the poster for Open Day	Kishan Narotam & Nitesh Nana	26 August 2019	In Progress
Generate all test cases	Kishan Narotam & Nitesh Nana	26 August 2019	In Progress

G2 Telecommunications Group Meeting Minutes

The weekly Telecommunication Group Meetings' recorded minutes can be found from the next page onwards.

01: MEETING MINUTES: Housekeeping and introduction to projects

Meeting: **Weekly Supervisor Meeting 1**

Date: **Tuesday, 16th July 2019**

Time: **2:00 – 2:30 pm**

Location: **Convergence Lab, 1st Floor Chamber of Mines Building**

Supervisors Present: Prof L. Cheng, Dr O. Oyerinde, Mr. M. Cox, Mr. S. Achari, Mr Craig Carlson

Students Present: Massi Mapani, Thabiso Magwaza, Marissa van Wangaardt, Siyabonga Khoza, Wavhudi Tshithivhe, Tumisho Rachudu, Karabelo Maqabe, Kishan Narotam, Matthew van Rooyen, Nitesh Nana, Brendon Swanepoel, James Goodhead, Robert Bradfield, Lynch Mwaniki, Tokelo Nchabeleng, Xongile Nchatsane

1. All weekly meetings will occur on Tuesdays at Convergence Lab from 12pm to 1pm.
2. Groups will chair meetings and those chairing a meeting will nominate the group for the preceding week
3. Groups chairing the meeting are responsible for minutes and sending calendar invites for the meetings.
4. Meetings should be information sharing and encourage team work as projects are similar.
5. Concerns and alike should be communicated in the meetings.
6. Discussion of Projects, their progress and current concerns that the different groups are facing while tackling their projects was discussed

Supervisor	Students	ProjectID
Prof. F. Takawira	Lynch Mwaniki and Tokelo Nchabeleng	19P63
	Kishan Narotam and Nitesh Nana	19P64
Dr Jules Moualeu	Siyabonga Khoza and Wavhudi Tshithivhe	19P55
Shamin Achari	Brendon Swanepoel and James Goodhead	19P34
Prof Ling Cheng	Massi Mapani and Thabiso Magwaza	19P05
Mr Craig Carlson	Tumisho Rachuene and Karabelo Maqabe	19P27
Dr Olutayo Oyerinde	Robert Bradfield and Matthew Van Rooyen	19P29
Mr. M. Cox	Marissa Van Wyngaardt and Xongile Nghatsane	19P12

02: MEETING MINUTES: Major Problem Discussions

Meeting: **Weekly Supervisor Meeting 2**

Date: **Tuesday, 23th July 2019**

Time: **12:00 – 13:00 pm**

Location: **Convergence Lab, 1st Floor Chamber of Mines Building**

Supervisors Present: Dr O. Oyerinde, Mr. M. Cox, Mr. S. Achari, Dr F. Takawira

Students Present: Massi Mapani, Thabiso Magwaza, Marissa van Wangaardt, Siyabonga Khoza, Wavhudi Tshithivhe, Tumisho Rachudu, Karabelo Maqabe, Kishan Narotam, Matthew van Rooyen, Nitesh Nana, Brendon Swanepoel, James Goodhead, Robert Bradfield, Lynch Mwaniki, Tokelo Nchabeleng, Xongile Nchatsane, Gift Langa, Mokulwane Nchabeleng

1. Weekly meetings to be moved to a more formal venue with Prof. Takawira suggesting a conference room. Destination change will be followed up on by next week's chair group.
2. Meeting minutes will be typed up by the current chair group allocated to that given week.
3. Discussions into the biggest problems faced by various groups during the last week with possible solutions given out by both the supervisors and groups who have had similar issues or whose project falls within the same field. These discussions have been summarized in the tables below
4. Group 1 (Kishan & Nitesh) were selected to chair next week's meeting.
5. Meeting was adjourned at 13h13.

Group Number:	19G41
Group Members:	Matthew van Rooyen (706692) Robert Bradfield (675515)
Problem(s) given:	Major bottleneck in the project's progression at this stage is the lack of a fruit database with which to train the machine learning model. Limited resources detailing tomato fruit diseases within data sets.
Possible solutions provided:	Possible solution was to change the scope of the project to a tomato grading system. It was decided that this differed too far from the original project specifications and would not be acceptable.
Further comments:	Discussions were undertaken as to the amount of image processing being performed during the project to ensure it met the necessary engineering requirements.

Group Number:	19G28
Group Members:	Mokulwane Nchabeleng and Lynch Mwaniki
Problem(s) given:	Difficulty training the Neural Network with compatible hardware. TensorFlow 1.14 requires CPUs to support the AVX instructions set. Group members laptops do not support the current TensorFlow version. Having trouble receiving a QAM signal wirelessly on the SDR.
Possible solutions provided:	Ask for PC from D-Lab that could be used on Demonstration day to run inference with the Neural Network. The neural network could be trained online using Kaggle or Google Colaboratory. Alternatively look into using MATLAB to create the Neural Network. Professor Takawira suggested we try increase the Transmitter power on the SDR. .
Further comments:	Professor Takawira stressed that Feature Extraction is important for those doing classification projects.

Group Number:	19G21

Group Members:	Karabelo Maqabe 1149346 Tumisho Rachuene 1314101
Problem(s) given:	Reading in values from the receiver side loses sync after approximately 10 readings and starts to cause all readings after that to lose sync and ignore start and end markers. This means the buffer is either being read too fast or slow at some point. Another problem is to set a reference point for clean ai of the sensors.
Possible solutions provided:	Increase the baude rate to 115200 and make sure the baude rate of the Bluetooth and the microcontroller are the same. As long as you can not smell smoke or gas you can consider the air to be clean and that value as a reference point.
Further comments:	We have created a method to separate the Carbon monoxide sensed in the smoke by calibrating the sensor to be sensitive to CO with the use of the sensitivity curves of the sensors in the data sheet.

Group Number:	19G01
Group Members:	Kishan Narotam (717 931) Nitesh Nana (720 064)
Problem(s) given:	· Not many problems experienced · Minor tweaking required in maths for DCT compression

Possible solutions provided:	· Minor tweaking required for DCT, more mathematical than anything else.
Further comments:	Project is on course. New ideas and possibilities are being realised and jotted down. The ult imate goal is the same, and is not changing, extra material is being thought of for open day.

Group Number:	19G23
Group Members:	Wavhudi Tshithivhe and Siyabonga Khoza
Problem(s) given:	The group had challenges with wireless transmission using the USRP1's, the USRP's failed to transmit a simple test signal wirelessly. This affected our approach in the project as most of the research focuses on using the usrp's for implementing wireless cooperative communication system.
Possible solutions provided:	Possible solution was to use the USRP 2's which were also not available. Another possible solution is to make use of Arduino with NRF24L01 transceiver to attain wireless communication.
Further comments:	Struggling to find an effective way of implementing cooperative communication system, as most of the implementation of system depend on USRP's for functionality and testing.

Group Number:	19G08
Group Members:	Brendon Swanepoel, James Goodhead
Problem(s) given:	There is a buffer overflow happening somewhere in the code and we are not sure where it is. This is stopping large amounts of data being sent via the serial connection.

Possible solutions provided:	Going to try update the data structures used and run the code in parallel in order to isolate the displaying of the frames and the reading of the frames from the serial connection.

Group Number:	19G14
Group Members:	Massi Mapani & Thabiso Magwaza
Problem(s) given:	Classifier built for experimental set up of identification of objects from office wall does not have a diversified dataset to train with; hence the testing and classification is not yielding expected results
Possible solutions provided:	Create a diversified dataset for better transfer learning of the classifier. Ensure to keep in mind the engineering input for the system also needs to be identified for the project that upgrades it from a weekend project to a 4th year engineering project.

03: MEETING MINUTES : Update from Current problems experienced and attempts made to rectify

Meeting: **Weekly Supervisor Meeting 3**

Date: **Tuesday, 30th July 2019**

Time: **12:00 – 13:00 pm**

Location: **Convergence Lab, 1st Floor Chamber of Mines Building**

Supervisors Present: Dr O. Oyerinde, Mr. M. Cox, Mr. S. Achari, Dr L. Cheng, Dr Jules Moualeu

Students Present: Massi Mapani, Thabiso Magwaza, Marissa van Wangaardt,, Wavhudi Tshithivhe, Tumisho Rachudu, Karabelo Maqabe, Kishan Narotam, Brendon Swanepoel, James Goodhead, Robert Bradfield, Lynch Mwaniki, Tokelo Nchabeleng, Xongile Nchatsane, Mokulwane Nchabeleng, Malebo Maboko

Students Absent: Nitesh Nana (Funeral in the family), Matthew van Rooyen (Work conflict), Siyabonga Khoza (Sick), Gift Langa (Work conflict, traffic caused delay)

1. Updates given on the problems discussed and mentioned from last week.
2. Any new issues or problems encountered must be stated and as the previous week, discussed with the group. Possible solutions will be given and alternatives may be stated.
3. The project is halfway through completion and by this time, projects should be reaching the end stages, and groups should start getting results for the presentation and begin working on the report.
4. Marissa and Xongile have been nominated to chair next week's meeting.
5. The venue will not change from Convergence Lab
6. The meeting was adjourned at 12:53

Group Number	19G41
Group Members	Matthew van Rooyen (706692) Robert Bradfield (675515)
Update from last week's problems	• Finding a data set of appropriate size and quality. • Wits agricultural department is the most promising department

Group Number	19G28
Group Members	Mokulwane Nchabeleng and Lynch Mwaniki
Update from last week's problems	The equalizer was not allowing the QAM to be seen properly.

Possible solutions given	Work on the custom block so that they can integrate the neural network.

Group Number	19G21
Group Members	Karabelo Maqabe (1149346) Tumisho Rachuene (1314101)
Update from last week's problems	The synchronicity was fixed by introducing a delay in the transmitter since the values were being sent faster than the receiver can process the values.
New problems/issues encountered	Using log function for converting ADC values to parts per million with a microcontroller causes the system to slow down.
Possible solutions given	It was suggested we instead use a linear approximated graph to perform the conversion to ppm instead of a log function or use a look up table which will not require too much processing from the microcontroller.
Further comments (are you on track with the project, behind, ahead of schedule?)	The project is on track and we are planning to move the system to a vero-board by the end of this week and perform testing.

Group Number	19G01
Group Members	Kishan Narotam & Nitesh Nana
Update from last week's problems	The DCT concern from last week has been resolved by consulting our supervisor. The DCT can be left as a high quality and low compression.
New problems/issues encountered	The run length encoding, needs very fine tweaking. Certain pixels are not being recorded.
Possible solutions given	Minor tweaking and maths possibly the issue or the use of MATLAB's built in function.

Further comments (are you on track with the project, behind, ahead of schedule?)	We are slightly ahead of the planned schedule, looking to be on track if the run-length encoding still poses an issue.

Group Number	19G23
Group Members	Wavhudi Tshithivhe & Siyabonga Khoza
Update from last week's problems	We have decided to go with using nrf24l01 combined with Arduino uno to create the wireless network and we have been able to transmit data between nodes.
New problems/issues encountered	We are currently encoding using hamming code on Matlab and are unable to write and read data from a file on Arduino software.
Possible solutions given	We will use CoolTerm to write data from Arduino to a text file. With reading data from a text file we have two possible solutions either we do hamming do hamming code on Arduino or we try find ways to receive and transmit data using Matlab
Further comments (are you on track with the project, behind, ahead of schedule?)	We are behind schedule, as we are supposed to be starting with testing .

Group Number	19G08
Group Members	Brendon Swanepoel, James Goodhead
Update from last week's problems	The buffer problem that was being experienced was solved by restructuring the data structures used. Video stream across a physical cable was achieved.
New problems/issues encountered	When transmitting over the LED, a visible flashing is observed. Also received signal is not as clean as it should be.

Possible solutions given	To remove the flashing, the sent data needs to be encoded to maintain a constant output power. The received signal can be cleaned up by getting better components as well as filtering and other techniques.
Further comments (are you on track with the project, behind, ahead of schedule?)	We are slightly ahead of schedule at the moment but the hardware issues need to be sorted soon in order to have enough time to build a test bed and obtain the required results.

Group Number	19G14
Group Members	Massi Mapani and Thabiso Magwaza
Update from last week's problems	Given that the built set up was very sensitive to set up noise. This was solved through brute force creation of a noisy dataset to and training of the classifier with it. This performed well with 93% accuracy, however other approaches require investigation.
New problems/issues encountered	The system is not robust for set up noise and spatial difference hence investigation into elegant techniques such as edge detection, DCT and other image preprocessing steps is required.
Possible solutions given	Increasing the class of sampling from 3 to one with an overall low probability value to measure significant and valid improvements and investigation into other methods that make the system robust and unaffected by the noise and position components.
Further comments (are you on track with the project, behind, ahead of schedule?)	Substantial direction towards the goal has been attained so far

Group Number	19G18
Group Members	Marissa Van Wyngaardt and Xongile Nghatsane
Update from last week's problems	• Problems with the actual software • Used to the software itself, • Able to transmit through a cable
New problems/issues encountered	• Find a way to frame the data

Group Number	19G13
Group Members	Gift Langa Malebo Maboko
Update from last week's problems	Last week's problems are still a work in progress.
New problems/issues encountered	The problem crashed when more than ten UE's were used.
Possible solutions given	USe any available equipment to ensure the success of the simulations.
Further comments (are you on track with the project, behind, ahead of schedule?)	We are behind by one week

<u>04: MEETING MINUTES : Update from Current problems experienced and attempts made to rectify</u>

Meeting: **Weekly Supervisor Meeting 4**

Date: **Tuesday, 6th August 2019**

Time: **12:00 – 13:00 pm**

Location: **Convergence Lab, 1st Floor Chamber of Mines Building**

Supervisors Present: Dr O. Oyerinde, Mr. S. Achari, Dr F. Takawira, Mr Craig Carlson

Students Present: Massi Mapani, Thabiso Magwaza, Marissa van Wangaardt, Wavhudi Tshithivhe, Tumisho Rachudu, Karabelo Maqabe, Kishan Narotam, Brendon Swanepoel, Robert Bradfield, Lynch Mwaniki, Tokelo Nchabeleng, Xongile Nchatsane, Mokulwane Nchabeleng, Malebo Maboko

Students Absent: James Goodhead (Printing PCB for project)

1. Updates given on the problems discussed and mentioned from last week.
2. Any new issues or problems encountered must be stated and as the previous week, discussed with the group. Possible solutions will be given and alternatives may be stated.
3. Groups should have started getting results for the presentation and begun working on their reports.
4. Brendon and James have been nominated to chair next week's meeting.
5. A suggestion was made to identify groups concerned with the progress of their project and focus the remaining meetings on solving issues in these projects.
6. The meeting was adjourned at 12:42.

Group Number	19G41
Group Members	Matthew van Rooyen (706692) and Robert Bradfield (675515)
Update from last week's problems	● Last week was dedicated to looking at online resources and developing own database manually
New problems/issues encountered	● Concerned that there will not be enough training data ● Shifting focus to code optimization, simplifying the system by reducing the number of functions and layers used in order to run it on less powerful processors ● Attempting to make the system work in real-time

Group Number	19G28
Group Members	Mokulwane Nchabeleng & Lynch Mwaniki

Update from last week's problems	● Spent time trying to reduce the number of inputs needed.
New problems/issues encountered	● Currently building custom block to perform feature extraction.
Possible solutions given	● Suggestion was made to explain the selection of values for neural network, even if selected experimentally.

Group Number	19G21
Group Members	Karabelo Maqabe (1149346) & Tumisho Rachuene (1314101)
Update from last week's problems	● Resolved buffer loss issues ● Made use of suggested straight-line approximation for conversion to ppm - losing too much accuracy
New problems/issues encountered	● Currently trying to compensate for temperature drift of sensor ● Using lookup table instead of straight line or computationally expensive log graphs

Group Number	19G01
Group Members	Kishan Narotam & Nitesh Nana
Update from last week's problems	● Resolved problem of run length encoding algorithm overlooking certain values.
New problems/issues encountered	● Going to introduce errors to the data to test algorithm's recovery ability ● Concerned about plagiarism/copyright infringement as algorithm was adapted for this project from an existing work
Possible solutions given	● Credit the source of the original algorithm

Group Number	19G23
Group Members	Wavhudi Tshithivhe & Siyabonga Khoza
Update from last week’s problems	● Having difficulty retrieving MATLAB data, difficulties working with binary data
New problems/issues encountered	● Able to transmit simple strings - will be generating random data and forming a black and white image from this as it will make it easier to spot transmission errors (using MATLAB’s binary to pixel functionality)

Group Number	19G08
Group Members	Brendon Swanepoel & James Goodhead
Update from last week’s problems	● Able to transit data but LEDs have visible flickering caused by a problem writing to the on-chip buffer.
New problems/issues encountered	● Attempting to transmit from buffer in 100-frame chunks on separate threads.
Possible solutions given	● Transmitting at higher data rates seems to remove flicker, will be moving circuits to PCB to improve switching times

Group Number	19G14
Group Members	Massi Mapani & Thabiso Magwaza
Update from last week’s problems	● Diversified classes in order to tell if accuracy is improving
New problems/issues encountered	● Moving from MATLAB to Google Collab to use GPUs provided ● Looking into custom image processor to do DCT pre-processing

Group Number	19G18
Group Members	Marissa Van Wyngaardt & Xongile Nghatsane
Update from last week's problems	● Completing the implementation of OOK modulation, will be testing this scheme
New problems/issues encountered	● Unsure how to quantify SNR ● Currently using the lower-frequency output of laser driver as it is less likely to damage the driver, will need to move to the higher threshold output in order to achieve faster data rates
Possible solutions given	● Characterise channel noise, attenuation ● Look at noise figure of detector

Group Number	19G13
Group Members	Gift Langa & Malebo Maboko
Update from last week's problems	● Still trying to reproduce algorithms found in research papers ● Overcame device limit, can hand over more than 10 devices
New problems/issues encountered	● Debugging code this week ● Will be trying to develop own algorithm

05: MEETING MINUTES : Update from Current problems experienced and attempts made to rectify

Meeting: **Weekly Supervisor Meeting 5**

Date: **Tuesday, 13th August 2019**

Time: **12:00 – 13:00 pm**

Location: **Convergence Lab, 1st Floor Chamber of Mines Building**

Supervisors Present: Dr O. Oyerinde, Mr. M. Cox, Mr. S. Achari, Dr L. Cheng, Dr Jules Moualeu, Dr F. Takawira, Mr Craig Carlson

Students Present: Massi Mapani, Thabiso Magwaza, Marissa van Wangaardt, Tumisho Rachudu, Karabelo Maqabe, Kishan Narotam, Brendon Swanepoel, James Goodhead, Mathew van Rooyen, Robert Bradfield, Lynch Mwaniki, Tokelo Nchabeleng, Xongile Nchatsane, Siyabonga Khoza, Wavhudi Tshithivhe, Gift Langa, Nitesh Nana

Students Absent: Malebo Maboko, Mokulwane Nchabeleng,

1. There is a presentation being given for the Telecoms group which all students in the Telecoms project group are invited to. An RSVP to Thabiso is required.
2. The meeting format will be changed to allow more focus to be given to groups that are struggling with their projects.
3. Updates given on the problems discussed and mentioned from last week.
4. Groups should put a lot of thought into how they are going to present for the final examination.
5. The assessment uses a rubric, pay attention to the rubric because the examiners will ask questions based on the rubric.
6. There will be no meeting for the week of open day.
7. Gift and Malebo have been nominated to chair next week's meeting.
8. The meeting was adjourned at 12:45.

Group Number	19G41
Group Members	Matthew van Rooyen Robert Bradfield
Update from last week's problems	• Tomato Database had been created with sufficient depth in order to train a model • Training algorithm optimized

New problems/issues encountered	• Object Detector is far too slow when identifying images • Inability to acquire diseased plants for demonstration on open day.
Further comments (are you on track with the project, behind, ahead of schedule?)	Project is on schedule.

Group Number	19G01
Group Members	Kishan Narotam Nitesh Nana
Update from last week's problems	Run-length encoding algorithm that was being used did not work as expected. A new algorithm was found and implemented. Everything working as desired.
New problems/issues encountered	No new problems or issues, project is essentially complete
Possible solutions given	After discussion with supervisor, it's a matter of deciding how to publish the results.
Further comments (are you on track with the project, behind, ahead of schedule?)	Project ahead of schedule, minus results, essentially complete. No optimization desired.

Group Number	19G08
Group Members	Brendon Swanepoel

	James Goodhead
Update from last week’s problems	The flicker has been reduced by applying the multi threading yet it is still slightly visible.
New problems/issues encountered	The flicker is still an issue but not a major concern. Results are now to be obtained for analysis.
Possible solutions given	Take results from different distances to obtain a range of packet loss and SNR values for analysis.
Further comments (are you on track with the project, behind, ahead of schedule?)	Project is on schedule at the moment.

Group Number	19G13
Group Members	Gift Langa & Malebo Maboko
Update from last week’s problems	• Bug fixes • Parameter setting
New problems/issues encountered	• Generation of results for the existing Handover Algorithm. • Development of new handover algorithm.
Possible solutions given	Using Matlab to analyse simulation statistics.
Further comments (are you on track with the project, behind, ahead of schedule?)	We are behind schedule by a week.

Group Number	19G14
Group Members	Thabiso Magwaza and Massi Mapani
Update from last week’s problems	The identification of pre-processing algorithms for robustness to set-up noise. Class size was increased.
New problems/issues encountered	Implementation and acquiring of results of implementation to the results is to be done this week
Possible solutions given	Implementation required
Further comments (are you on track with the project, behind, ahead of schedule?)	Slightly behind running out of time

Group Number	19G18
Group Members	Marissa Van Wyngaardt & Xongile Nghatsane
Update from last week’s problems	• Making use of FFT plots to observe SNR when transmitting data
New problems/issues encountered	• First byte of packet header has errors due to differential decoding implementation clashing with GNURadio functionality • If header is error-free, payload usually is as well - difficult to estimate BER
Possible solutions given	• Discard broken packets, focus on substitution errors rather than synchronization errors • Increasing payload length will increase likelihood that errors will occur in payload

Further comments (are you on track with the project, behind, ahead of schedule?)	Behind schedule as we aren't testing system performance (SNR vs BER, etc) as of yet

Group Number	Group 21
Group Members	Karabelo Maqabe (1149346) Tumisho Rachuene (1314101)
Update from last week's problems	We calibrated the system manually with a reset button that a user can press when they believe they are in clean air which will be the reference point.
New problems/issues encountered	There were no problems, but we needed to test the system and realized we need to include a humidity and temperature sensors to be able to compare the system to existing quality monitoring systems.
Possible solutions given	None
Further comments (are you on track with the project, behind, ahead of schedule?)	Implement the sensor and perform system tests to obtain technical specifications of the device and the project is currently on track.

Group Number	19G23
Group Members	Siyabonga and Wavhudi
Update from last week's problems	Last week the group was struggling to transmit the data from Relay to Destination. The group managed

	to sort out the issue and now our system can transmit.
New problems/issues encountered	We are struggling to send a big pixel picture which is made up of a 2D binary array.
Possible solutions given	Divide the array into smaller packets (1D array) to ensure that encoding, decoding, transmission and receiving becomes manageable.
Further comments (are you on track with the project, behind, ahead of schedule?)	The group is behind with the project, we still need to test the system by introducing deep fading into system.

Group Number	19G28
Group Members	Lynch Mwaniki and Mokhulwane Nchabeleng
Update from last week's problems	Began attempt at creating embedded Python blocks in GNURadio that would do feature extraction and load the Neural Network model for classification.
New problems/issues encountered	Installing the necessary packages in GNURadio's pip python package manager such as TensorFlow has proven to be a challenge. The current TensorFlow version it has is outdated. Currently, looking for TensorFlow versions built for Python 2.7 on Windows.

Possible solutions given	Look for repos on GitHub that host Tensorflow packages built from source for Windows.
Further comments (are you on track with the project, behind, ahead of schedule?)	We are currently on schedule.

06: MEETING MINUTES : Update from Current problems experienced and attempts made to rectify

Meeting: **Weekly Supervisor Meeting 6**

Date: **Tuesday, 20th August 2019**

Time: **12:00 – 13:00 pm**

Location: **Convergence Lab, 1st Floor Chamber of Mines Building**

Supervisors Present: Mr. M. Cox, Mr. S. Achari, Dr L. Cheng

Students Present: Massi Mapani, Thabiso Magwaza, Marissa van Wangaardt, Tumisho Rachudu, Karabelo Maqabe, Kishan Narotam, Brendon Swanepoel, James Goodhead, Mathew van Rooyen, Robert Bradfield, Lynch Mwaniki, Tokelo Nchabeleng, Xongile Nchatsane, Siyabonga Khoza, Wavhudi Tshithivhe, Gift Langa, Nitesh Nana

Supervisors Absent: Dr O. Oyerinde, Dr Jules Moualeu, Dr F. Takawira, Mr Craig Carlson

1. Updates given on the problems discussed and mentioned from last week.
2. Lecturers gave advice about open day and the general content that should be in the poster.
3. The assessment uses a rubric, pay attention to the rubric because the examiners will ask questions based on the rubric.
4. There is a possible lecture about doing presentations that will take place on the week of presentations
5. There will be no meeting for the week of open day.

Group Number	19G13
Group Members	Gift Langa and Malebo Maboko
Update from last week's problems	• Testing of the proposed algorithm • Finalising existing algorithm
New problems/issues encountered	Simulation was crashing
Possible solutions given	Find optimal values to use in the simulator without it crashing
Further comments (are you on track with the project, behind, ahead of schedule?)	Generation of results of proposed algorithm and existing algorithm

Group Number	19G14
Group Members	Thabiso Magwaza and Massi Mapani
Update from last week's problems	The data set had to be updated to carry out the checks per algorithm
New problems/issues encountered	No problems encountered just time being the main concern
Possible solutions given	Work longer hours and ensure the classifier structure is optimal

Further comments (are you on track with the project, behind, ahead of schedule?)	Endeavouring to close the project off.

Group Number	19G23
Group Members	Siyabonga Khoza (894206) and Wavhudi Tshithivhe (1160095)
Update from last week's problems	We still experiencing the same problem as last week, we are unable to transmit and receive a big array of bits.
New problems/issues encountered	When sending multiple arrays we experiencing burst errors and we are unable to send a large array at once as our equipment, Arduino Uno does not have the capacity.
Possible solutions given	We will try and send smaller arrays with indexes to determine the order in which the arrays are sent at the destination.
Further comments (are you on track with the project, behind, ahead of schedule?)	Currently we are working on open day poster and what we will showcase to the people.

Group Number	19G01
Group Members	Kishan Narotam Nitesh Nana
Update from last week's problems	All problems resolved, closing up the project
New problems/issues encountered	Not so much a problem, but just a confirmation needed from the supervisor on whether the test cases we want to run.
Possible solutions given	N/A
Further comments (are you on track with the project, behind, ahead of schedule?)	On track, pressure is building in terms of attaining results and creating the poster.

Group Number	Group 21
Group Members	Karabelo Maqabe (1149346) Tumisho Rachuene (1314101)
Update from last week's problems	The temperature and humidity sensor was successfully implemented so that testing can begin.
New problems/issues encountered	We are manually calibrating the system and therefore can not know how accurate the system will be
Possible solutions given	Find out how much parts per million you are possibly losing by self-calibrating the system.

Further comments (are you on track with the project, behind, ahead of schedule?)	The project is on track. We are performing system tests this week and hopefully implement the interface for the device.

Group Number	19G28
Group Members	Mokhulwane Nchabeleng and Lynch Mwaniki
Update from last week's problems	We were able to import the Neural Network model into our Embedded Python blocks in GNURadio. It can classify QAM signals sent over a coaxial cable.
New problems/issues encountered	Trying to go wireless and transmit the signals using antennas. The current ISM bands seem to have interference thus making it difficult to receive the desired transmitted signal.
Possible solutions given	Check other ISM bands that may have less interference, but do not use non ISM bands as it would be illegal..
Further comments (are you on track with the project, behind, ahead of schedule?)	We are on track with the project.

Group Number	19G08
Group Members	Brendon Swanepoel James Goodhead
Update from last week's problems	Results for the system are being obtained currently.
New problems/issues encountered	Code for open day needs to be finalised.
Possible solutions given	Split the work between the partners to ensure that the results are obtained and the code is finished in time.

Further comments (are you on track with the project, behind, ahead of schedule?)	Project is on schedule at the moment.