JPEG COMPRESSION USING MATLAB

by

NAVNEETH S 19BEC1309

KEVIN GERAD THOMAS 19BEC1359

AKHIL K S 19BEC1417

A project report submitted to

**Dr. SUCHETHA M**

**SCHOOL OF ELECTRONICS ENGINEERING**

in partial fulfilment of the requirements for the course of

**ECE2006–DIGITAL SIGNAL PROCESSING**

in

**B.Tech. ELECTRONICS AND COMMUNICATION ENGINEERING**

****

**Vandalur – Kelambakkam Road**

**Chennai – 600127**

**NOVEMBER 2021**

**BONAFIDE CERTIFICATE**

Certified that this project report entitled “JPEG COMPRESSION USING MATLAB**”** is a bonafide work of **NAVNEETH S – 19BEC1309, KEVIN GERAD THOMAS – 19BEC1359** and **AKHIL K S - 19BEC1417** who carried out the Project work under my supervision and guidance for **DIGITAL SIGNAL PROCESSING**

**Dr. SUCHETHA M**

Associate Professor Senior & HoD-B.Tech(ECE)

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

**ABSTRACT**

Creating, editing, and generating images in a very regular system today is a major priority. The original image data generated by the camera sensor is very large to store, so the efficiency is not high. Mobile or bandwidth-limited systems become particularly cumbersome, where the object is a conservative bandwidth cost, such as the World Wide Web. This situation requires the use of efficient image compression techniques, such as JPEG algorithm techniques, that perceive images with almost no loss of compressed image height.

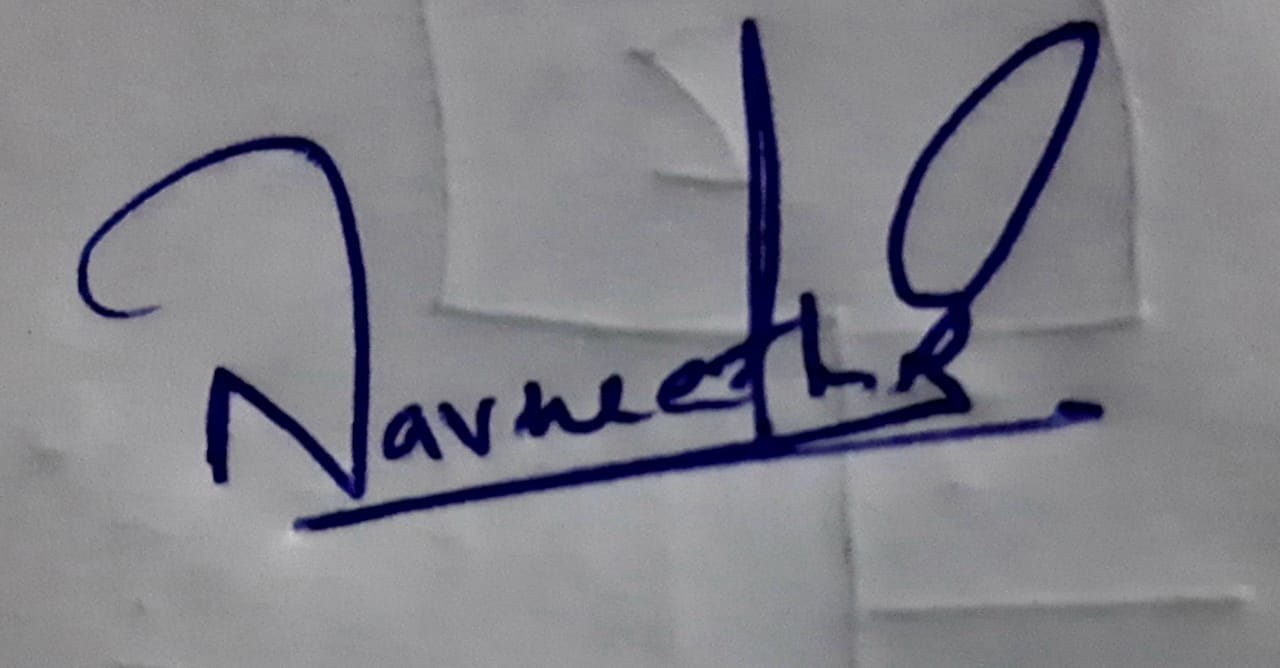
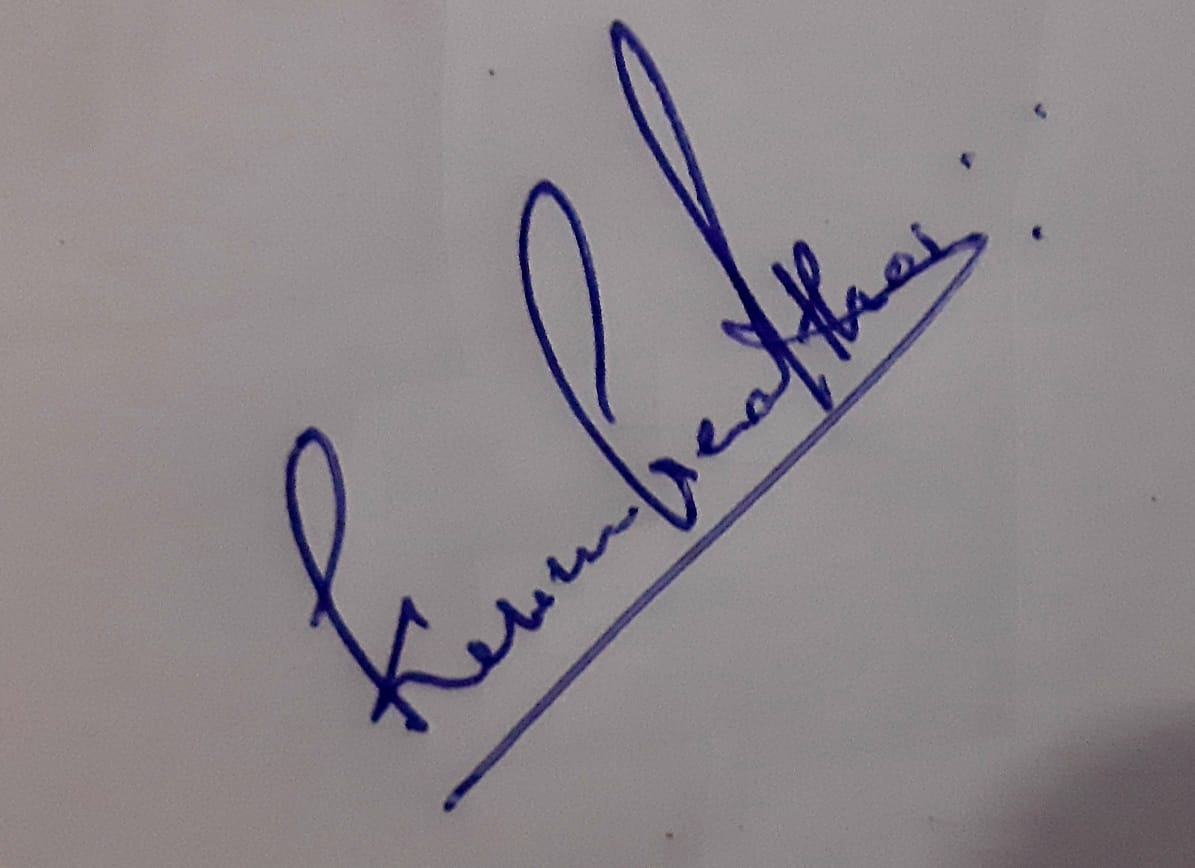
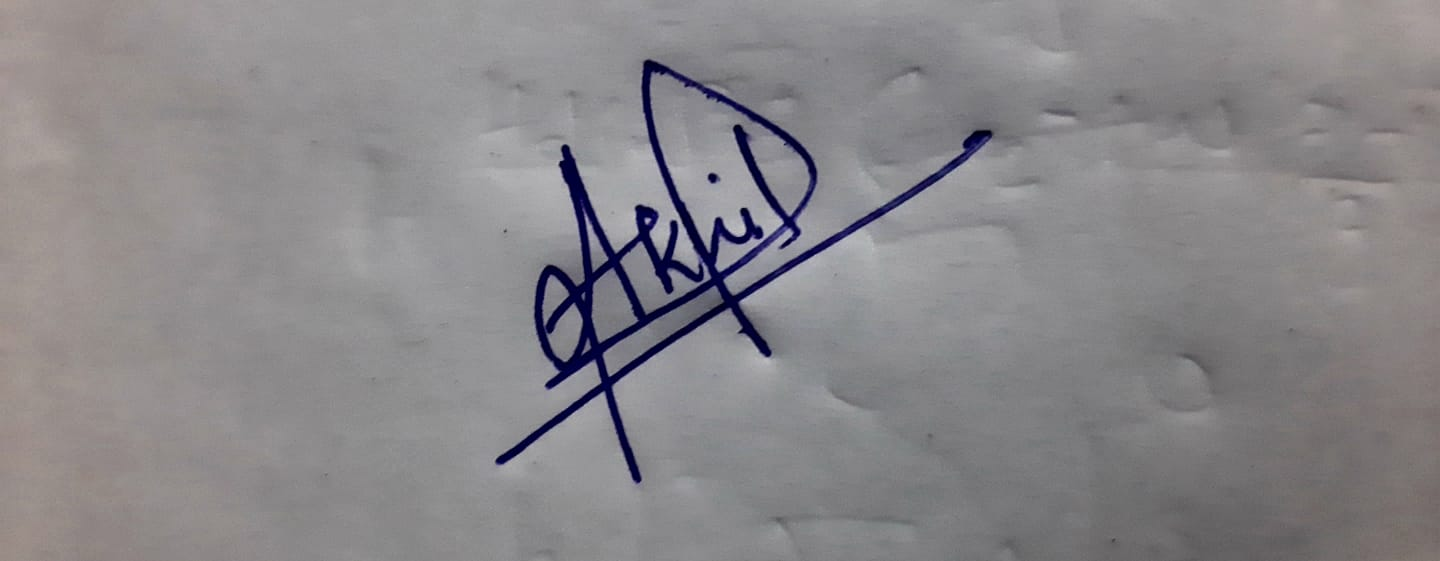
**ACKNOWLEDGEMENT**

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Suchetha M,** Associate Professor Senior, School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Sivasubramanian. A,** Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

  ****

**NAVNEETH S KEVIN AKHIL K S**

**TABLE OF CONTENTS**

| **SL NO** |  | **INDEX** | **PAGE** |
| --- | --- | --- | --- |
|  |  | ABSTRACT | 3 |
|  |  | ACKNOWLEDGEMENT | 4 |
| **1** |  | INTRODUCTION | 6 |
|  | **1.1** | OBJECTIVE | 6 |
|  |  |  |  |
| **2** |  | DISCRETE COSINE TRANSFORM(DCT)- DESIGN | **7** |
|  | **2.1** | BLOCK DIAGRAM | **7** |
|  | **2.1.1** | FORMULA | **8** |
|  | **2.2** | PROCESS | **8** |
|  | **2.2.1** | DCT ALGORITHM | **9** |
|  | **2.3** | SSIM: Structural Similarity Index | **9** |
| **3** |  | IMPLEMENTATION OF CODE | **10** |
|  | **3.1** | CODE IMPLEMENTATION | **10-13** |
|  | **3.2** | SAMPLES OF COMPRESSED IMAGES | **13-15** |
|  | **3.3** | MATLAB CODE FOR SSIM | **16** |
|  | **3.3.1** | COMPRESSED IMAGE | **16-20** |
|  | **3.4** | RESULT AND INFERENCE | **21** |
|  | **3.5** | APPLICATION | **21** |
| **4** | **4.1** | CONCLUSION | **21** |
|  | **4.2** | REFERENCE | **22** |
|  |  |  |  |

**CHAPTER-1**

**INTRODUCTION**

**1 INTRODUCTION**

* Transform coding constitutes an integral part of contemporary image / video processing applications.
* The transform coding relies on the premise that the pixels in the image exhibit a degree of correlation with their neighboring pixels. Similarly, in the video transmission system, these correlations can be used to predict the values of pixels from their neighbors.
* Thus, the transformation is defined to map the spatial (correlation) data into transform (uncorrelated) coefficients. The conversion should take advantage of the fact that the information content of a single pixel is relatively small, that is, to a large extent, its neighbors can be used to predict the visual contribution of the pixel

**1.1 OBJECTIVES**

* To design a matlab program to perform jpeg compression
* To understand the discrete cosine transform(DCT) in image compression

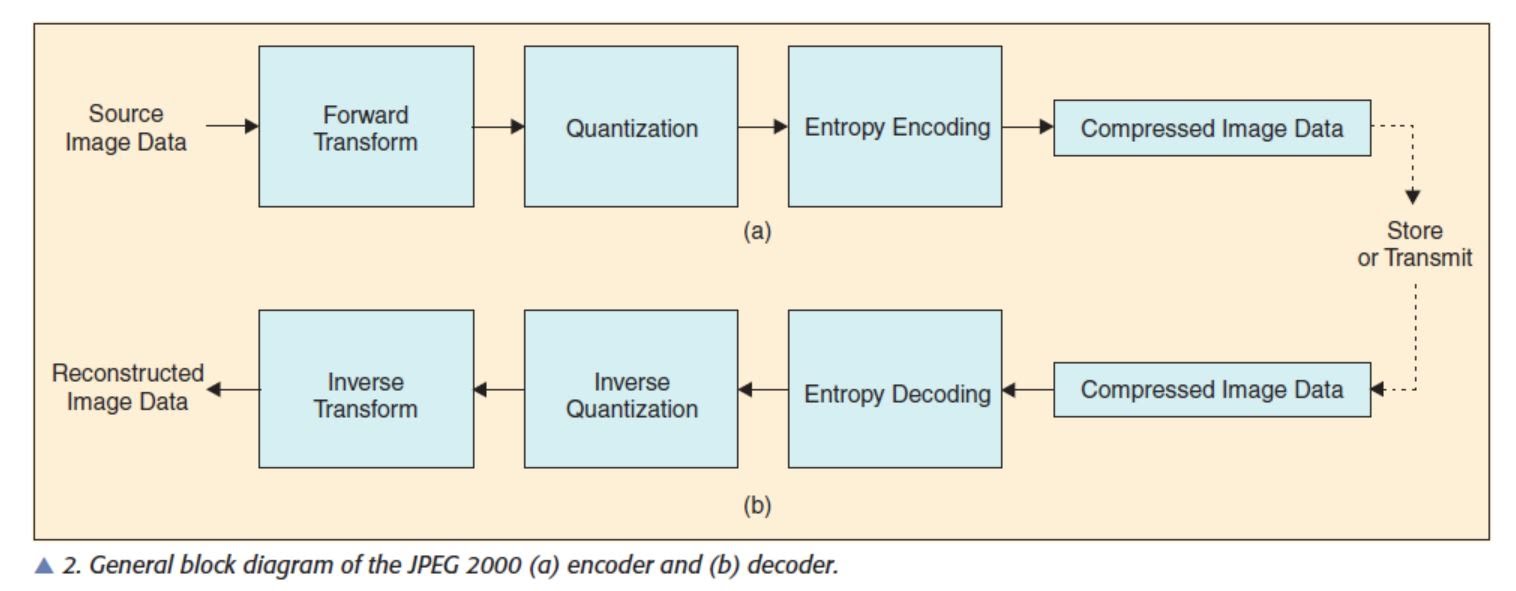
**CHAPTER- 2**

**DISCRETE COSINE TRANSFORM(DCT)- DESIGN**

**DISCRETE COSINE TRANSFORM(DCT)**

* The DCT is a technique allowing the conversion of a signal into elementary frequency components.
* In DCT, the input signal is represented as a linear combination of functions that are related to its frequency components.
* DCT does not directly reduce the number of bits required to represent the block. For instance, for an 8 × 8 block of 8-bit pixels, the DCT produces an 8 × 8 of 11-bit coefficients due to the range of coefficient values.
* However, considering that the DCT concentrate, the low-frequency coefficients, and remaining other coefficients are mainly zero, the compression can be achieved by transmitting the near-zero coefficients and by quantizing and coding the remaining coefficients.

**2.1 BLOCK DIAGRAM**

****

**2.1.1 FORMULA**

**Text, letter

Description automatically generated**

* P(x, y) is the x,yth element of the image represented by the matrix P
* N is the size of theblock that the DCT is done on.
* N is the size of the block that the DCT is done on.
* N equals 8 and x and y range from 0 to 7.

**2.2 PROCESS**

* A matrix of image is taken and is leveled off by subtracting 128 from each entry , m = is the image matrix
* The discrete cosine transform is performed on the new image matrix by matrix multiplication , d =tmt'.
* Compression by quantization , from range 1 to 100 , 1 is the poorest quality , for example 50 is selected.
* Quantization is done by each element in the transform matrix d by q(quantization matrix) then rounding it off to the nearest integer value.
* C(i,j) = round(Di,j/Qi,j)
* We will use the non-zero coefficients to reconstruct the image .
* Coding takes place by using encoder; all the coefficients are converted into binary streams .
* Then, for reconstruction the bit stream is then decodes , we get quantized matrix C , which is then by quantization matrix originally used here:
* Ri,j = Qi,j \* Ci,j
* Then idct is applied to matrix R , then is rounded to the nearest integer and 128 is added to each element of the result , N=round(T' R T)+128 , hence we get the final decompressed image.

**2.2.1 DCT ALGORITHM**

* The image is broken into 8×8 blocks of pixels.
* Working from left to right, top to bottom, the DCT is applied to each block.
* Each block is compressed through quantization.
* The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.
* When desired, the image is reconstructed through decompression, a process that uses the inverse Discrete Cosine Transform (IDCT).

**2.3 SSIM: Structural Similarity Index**

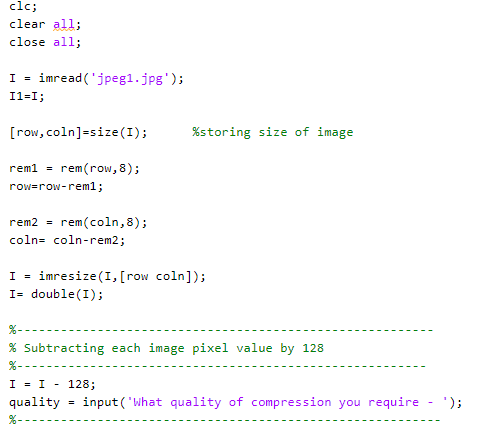
The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission. It is a full reference metric that requires two images from the same image capture— a reference image and a processed image. The processed image is typically compressed. It may, for example, be obtained by saving a reference image as a JPEG (at any quality level) then reading it back in.

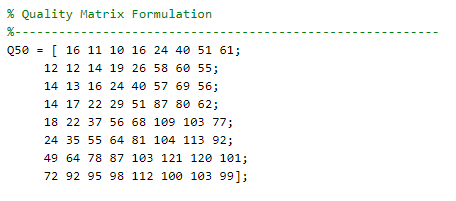
**CHAPTER - 3**

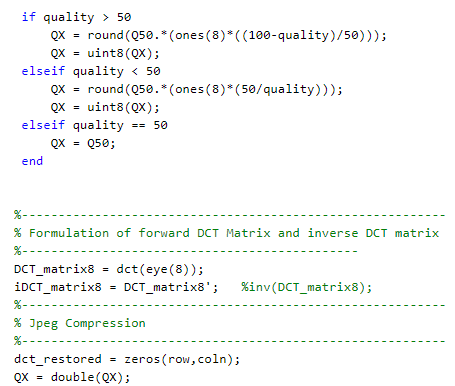
**IMPLEMENTATION OF MATLAB CODE**

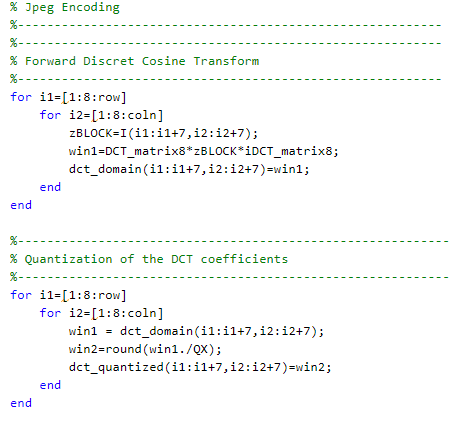
This section describes code implementation and results with inferences.

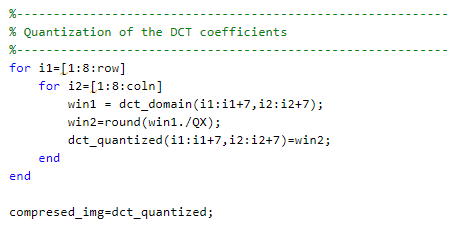
**3.1 CODE IMPLEMENTATION**

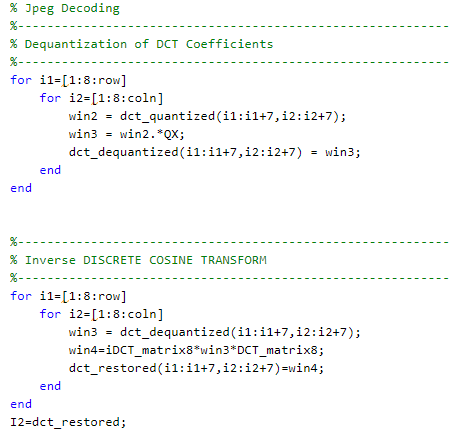
****

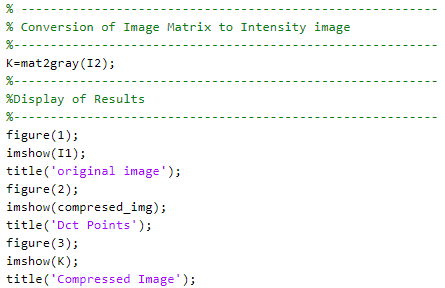
****

****

****

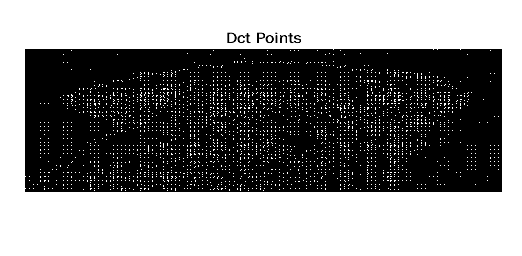
****

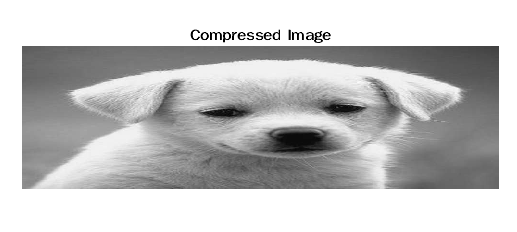
****

****

**3.2 SAMPLES OF COMPRESSED IMAGES AND THEIR DCT POINTS  
(COMPRESSION QUALITY-60)**

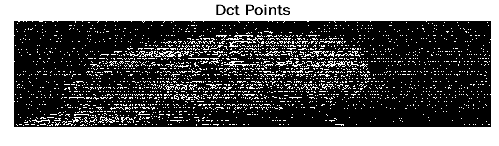






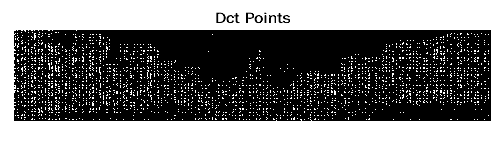
2





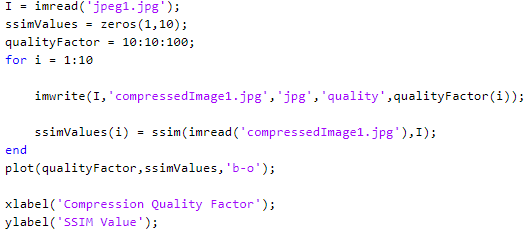
3





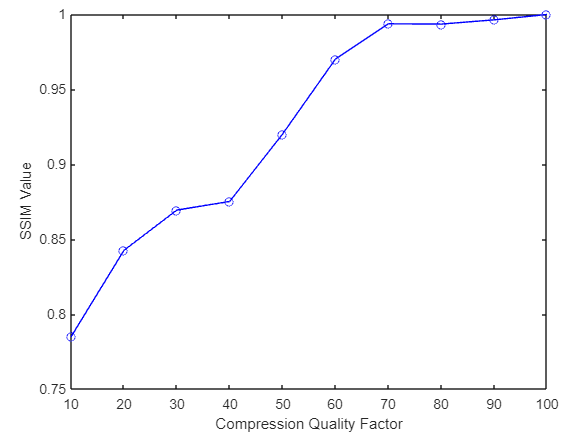
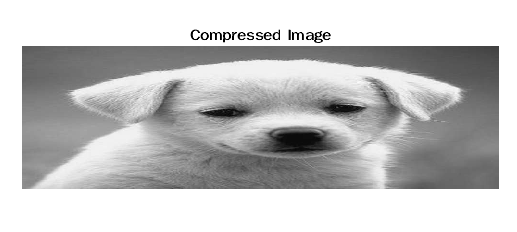


**3.3 MATLAB CODE FOR SSIM**

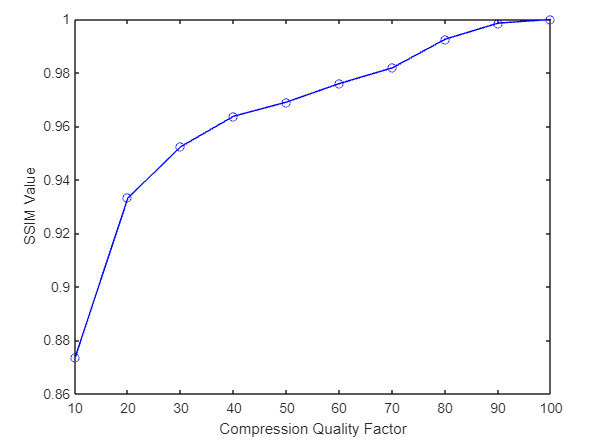
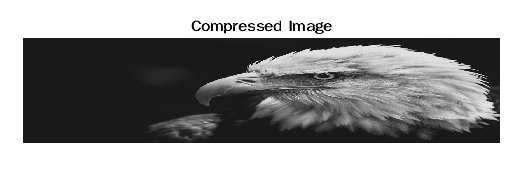
****

**3.3.1 COMPRESSED IMAGES AND THEIR SSIM PLOTS**

**1.**

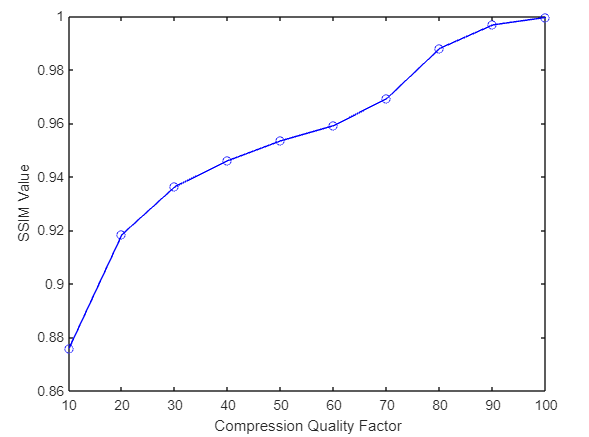
****

**2**

****

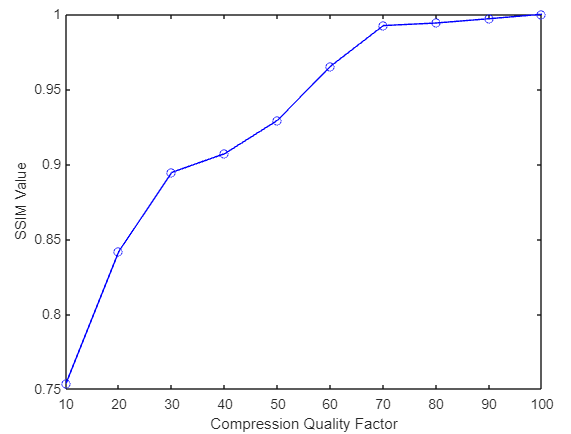
**3**

****

****

**4**

****

****

**3.4 RESULTS AND INFERENCES**

Was able to implement jpeg compression using dct in matlab successfully. We also analysed the quality of compression using ssim.

**3.5 Applications:**

· The images can be saved in the compressed format and can be reconverted to the RGB version when they must be displayed.

· The processed blocks of information can be sent over a communication channel, thus consuming less bandwidth.

· This processed DCT information can be given as input to Deep Learning based Computer Vision Tasks which usually need lots of high-quality data.

As a Data Scientist with a background in Electronics and Telecommunications which primarily focused on Signal Processing, I have seen and worked the first two parts mentioned in the applications section. My goal eventually is to implement the third part now.

References:

**CHAPTER - 4**

**4.1 CONCLUSION**

* Since jpeg is an image compression standard, this project studies the main process of jpeg encoding.
* Compression can be achieved by using DCT techniques that divide the image into different frequency components.
* The unnecessary information can then be removed from the image by quantization.
* This means that DCT plays a vital role in JPEG image compression. As the compression ratio increases, a huge loss of information is also observed. Therefore, the need to introduce a highly efficient DCT algorithm to achieve better image compression arises.

**4.2 REFERENCES**

1. Ahmed, N., T. Natarajan, and K. R. Rao. 1974. On image processing and a discrete cosinetransform. IEEE Transactions on Computers C-23(1): 90-93.
2. Chen, W. H., and W. K. Pratt. 1984. Scene adaptive coder. IEEE Transactions on Communications COM-32: 225-232.
3. Jain, A. K. 1989. Fundamentals of digital image processing . Prentice Hall: Englewood Cliffs,NJ.
4. Puri, A. 1992. Video coding using the MPEG-1 compression standard. Society for Information
5. Display Digest of Technical Papers 23: 123-126.

**BIODATA**



Name :

Mobile Number :

E-mail :

Permanent Address:



Name :

Mobile Number :

E-mail :

Permanent Address:

 Name :

Mobile Number :

E-mail :

Permanent Address:

