**Image Inpainting using KNN in DCT Domain and Compression to get an optimized output**

A Dissertation Submitted

**By**

**Biprajit Bhattacharjee**

**(11300364)**

to

**Department of Computer Science**

In partial fulfillment of the Requirement for the

Award of the Degree of

**Master of Technology in Computer Science & Engineering**

**Under the guidance of**

**Mr. Sanyam Anand**

**(May, 2015)**

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

School of: Computer Science & Engineering

**DISSERTATION TOPIC APPROVAL PERFORMA**

Name of the Student: Biprajit Bhattacharjee  Registration No: 11300364

Batch: 2013-15  Roll No.: B49

Session: 2014 15  Parent Section: K2306

Details of Supervisor:  Designation: AP

Name: SANYAM  Qualification: Mtech (IT)

U.ID: 15736  Research Experience: 2 years

SPECIALIZATION AREA: ARTIFICIAL INTELLIGENCE (pick from list of provided specialization areas by DAA)

PROPOSED TOPICS

1. Compression & decompression using SPIHT in Aerial application
2. Bee colony optimization of neural networks
3. Image den

Signature of Supervisor 15736

PAC Remarks:

Topic 1 is approved, Publication is also expected from the student.

APPROVAL OF PAC CHAIRPERSON: Signature: 11011  Date:

*Supervisor should finally encircle one topic out of three proposed topics and put up for approval before Project Approval Committee (PAC)

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to Supervisor.

## ABSTRACT

This report has major focus on images which have cracks and minor distortions in them. This prevents any compression technique to give the best output. Hence there is a need to first remove those distortions from the images before applying compression. A system is proposed, in which K-Nearest Neighbor Algorithm is implemented after the DCT Step, to remove the distortions before compressing it. The neighboring color of the distorted part can be used to merge with the damaged part. So basically, we have to store less color intensities to save the same image without the distorted part which will result in more optimized compression ratio. It will be greatly valuable for viewer perspective.

## ACKNOWLEDGEMENT

First of all I would like to thank my mentor, Mr. Sanyam Anand, Assistant Professor, Department of Computer Science Engineering, Lovely Professional University, Punjab for his valuable support and guidance during the entire course of work.

I would like to express my special thanks to our Digital Image Processing Teacher, Miss. Usha Mittal, Department of Computer Science Engineering, Lovely Professional University, Punjab for guiding me throughout the project work by giving me timely suggestions.

I would like to express my deep gratitude to all my batch mates and friends, whose valuable suggestions and support helped me to complete this.

Last but not the Least, I would like to thank to my parents for giving me their love and support throughout my life.

**Biprajit Bhattacharjee**

**Regd. No: 11300364**

## DECLARATION

I hereby declare that the dissertation entitled, **Image Inpainting using KNN in DCT Domain and Compression to get an optimized output** submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date: ________________

**Biprajit Bhattacharjee**

**Regd. No: 11300364**

# CERTIFICATE

This is to certify that **Biprajit Bhattacharjee** has completed M.Tech dissertation titled **Image Inpainting using KNN in DCT Domain and Compression to get an optimized output** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of the dissertation has ever been submitted for any other degree or diploma.

The dissertation is fit for the submission and the partial fulfillment of the conditions for the award of M.Tech Computer Science & Engg.

Date: ___________________

**Signature of Advisor**

**Name: Sanyam Anand**

**UID: 15736**

# List of Tables

**Table No** **Page No**

1.1 Comparison of Encoding Techniques 8
4.1: MSE and PSNR Values of the images from IVC Database 24
4.2: Compression Ratio and the Original and Compressed Size of the images from IVC Database before and after Inpainting 26
4.3: MSE and PSNR Values of the images from the Standard Image Database 28
4.4: Compression Ratio and the Original and Compressed Size of the images from Standard Image Database before and after Inpainting 29
4.5: MSE and PSNR values of the images from the Online database by Computer Vision 32
4.6: Compression Ratio and the Original and Compressed Size of the images from the Online database by Computer Vision 34
4.7: MSE and PSNR values of the images from the tid2008 database 36
4.8: Compression Ratio and the Original and Compressed Size of the images from the tid2008 database. 37
4.9: Individual R,G,B values of MSE and PSNR of the Color Images 41

# List of Figures

**Figure No** **Page No**


| Figure No | Page No |
|---|---|
| 1.1 Block Diagram of Lossy Algorithm | 3 |
| 1.2 Block Diagram of Lossless Algorithm | 3 |
| 1.3 Default Luminance Matrix | 7 |
| 1.4 Default Chrominance Matrix | 7 |
| 3.1 Proposed Methodology | 19 |
| 4.1: Images from the IVC Database of size 512x512 | 24 |
| 4.2: Original Image and its Compressed output of the images from the IVC Database | 26 |
| 4.3: Images from the Standard Image Database of size 512x512 | 28 |
| 4.4: Original Image and its Compressed output from the Standard Database. | 29 |
| 4.5: Few Images from the Online database by Computer Vision of Size 512x512 | 32 |
| 4.6: Original and its Compressed output of the images from the Online database by Computer Vision | 34 |
| 4.7: Few Images from the tid2008 database of variable sizes | 36 |
| 4.8: Original and its Compressed output of the images from the tid2008 database | 37 |
| 4.9: Example of Inpainting with Colored Images | 40 |
| 4.10: Example showing an image which won't work with the algorithm | 40 |
| 4.11: Example of Restored Images with their Zoomed View to show the restored pixels | 42 |

# Table of Contents

**Chapters** **Page Nos**


| Chapters | Page Nos |
|---|---|
| Abstract | i |
| Acknowledgment | ii |
| Declaration | iii |
| Certificate | iv |
| List of Tables | v |
| List of Figures | vi |
| **Chapter 1** | |
| Introduction | 1 |
| 1.1 Compression Algorithm | 2 |
| 1.1.1 Lossy Algorithm | 2 |
| 1.1.2 Losseless Algorithm | 3 |
| 1.2 Joint Photographic Experts Group (JPEG) | 3 |
| 1.3 Color Specification | 4 |
| 1.4 FDCT and IDCT | 5 |
| 1.5 Quantization | 6 |
| 1.6 Encoder | 7 |
| 1.7 Comparison of Encoding Techniques | 8 |
| 1.8 Image Inpainting/Restoration Techniques | 8 |
| 1.8.1 Image Restoration with K>3 | 9 |
| **Chapter 2** | |
| Literature Review | 11 |
| **Chapter 3** | |
| Present Work | 17 |
| 3.1 Problem formulation | 17 |
| 3.2 Objectives | 18 |

3.3 Methodology 18
3.3.1 Proposed Approach 18
3.3.2 Work Methodology 19
**Chapter 4**
Results and Discussion 20
4.1 Image Quality Evaluation 20
4.1.1 MSE and PSNR 20
4.1.2 Compression Ratio 21
4.2 Experiment Results 21
4.2.1 Experiments with grayscale images 21
4.2.2 Experiments with Colored Images (RGB Images) 38
4.3 Limitations of the algorithm 41

**Chapter 5**
Conclusion 43
5.1 Future Scope 44
**Chapter 6:** References 45
**Chapter 7**
Appendix 48
**Chapter 8**
Publications 49

# CHAPTER 1
# INTRODUCTION

At the present time where bulk of data is getting transferred each and every second, different compression methods is a necessity. By being able to compress images to a small fraction of their original size, we can save a significant amount of disk space. Likewise transportation of images starting from one computer then onto the next gets to be simpler and quick.

Image compression Algorithms can be categorized into: Lossy and Lossless. As the name states, decompression after lossless compression yields the same input image. Lossy compression brings about loss of information and the decompressed image is not precisely the same as the original one.

The Image inpainting alludes to the issue of filling-in missing regions in an image. Inpainting images with impediment or corruption is a challenging task. Most existing algorithms are pixel based, which develop a statistical model from image characteristics. One of the primary burdens of these methodologies is that, their viability is constrained by the surrounding pixels of the destroyed part. Subsequently, good performance of these strategies is acquired just when the images have particular consistency. Images in the frequency domain contain sufficient data for image inpainting and can be utilized as a part of data recreation, e.g., high frequency indicates image edges or textures, which motivates conducting image inpainting in the frequency domain. In this report, we propose a novel KNN method which will utilize the DCT coefficients in the frequency domain to remove the distortions. We look for an adequate representation for the functions and utilize the DCT coefficients of this representation to produce an over-complete dictionary. One considers exchanging the pixel-level processing method to the frequency domain for image inpainting utilizing discrete cosine change (DCT).

Existing strategies can be ordered into the accompanying classifications. The principal class concerns diffusion-based methodologies which proliferate level lines (called isophotes) utilizing partial differential equations (PDE) or variational systems. The second classification concerns examplar-based inpainting methods which have been motivated from texture synthesis procedures. These methods exploit image statistical and equivalence toward oneself priors. The texture to be incorporated is adapted by sampling, replicating or by edging patches (called examplar). These techniques have developed over recent years with the acquaintance of variations related with the patch processing order, to quick search of comparative patches, or to the presentation of spatial coherence constraints. An alternate class of methodologies concerns techniques utilizing sparsity priors.

### 1.1 Compression algorithm

As stated above, Image compression Algorithms can be partitioned into: Lossy and Lossless. Decompression after lossless compression yields the same input image. Lossy compression brings about loss of information and the decompressed image is not precisely the same as the first.

#### 1.1.1 Lossy Algorithm

This data compression technique is finished with loss of value and inventiveness (Fig. 1.1). The original information is never recovered over after compression. A couple of normal case of lossy compression is found in streaming features in the web, additionally the VOIP utilizes lossy compression strategy. The algorithm may pick a more modest extent of pixels whose color value contrasts fall inside the limits of our discernment, and substitute those for the others. The unwanted pixels can then be disposed of. Extremely noteworthy decreases in data size may be attained with this manifestation of compression, however the refinement of the algorithm decides the visual nature of the completed item.

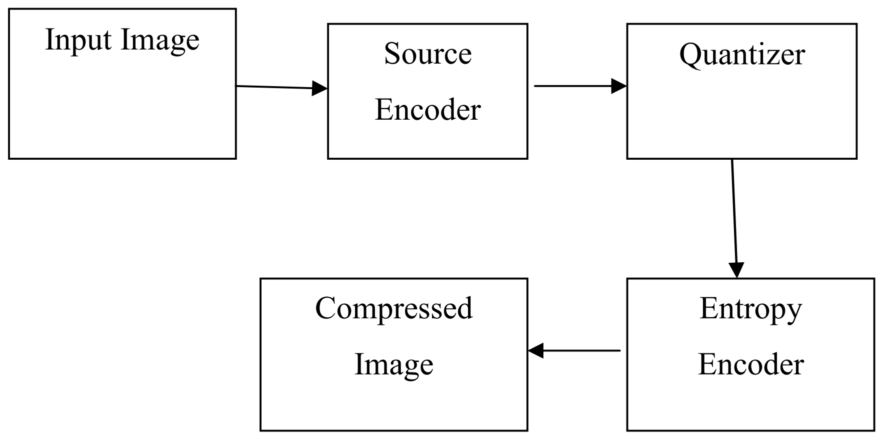



Fig1.1: Block Diagram of Lossy Algorithm

**1.1.2 Lossless Algorithm**

The data compression technique retains the originality of the file after compression by exploiting only the statistical redundancy of a data file (Fig. 1.2). The original file can be retrieved back after decompression. It can attain an unobtrusive measure of compression. It can minimize it to about half of that size, contingent upon the sort of file being compressed. This makes it helpful for exchanging files over the Internet, as smaller records exchange faster.

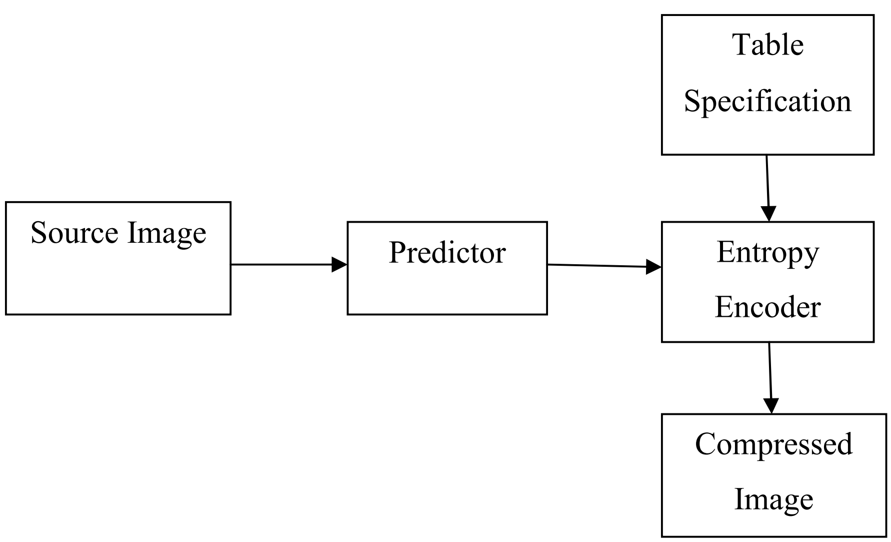



Fig 1.2: Block Diagram of Lossless Algorithm

## 1.2 Joint Photographic Experts Group (JPEG)

JPEG is an abbreviation for "Joint Photographic Experts Group". For its standard coding technique the Discrete Cosine Transform (DCT) is utilized. In light of the fact that each coefficient can be dealt autonomously without losing compression proficiency, de-correlation is essential for compression. Basically, the 2-D version of DCT disintegrates a nxn block of an image into a set, each with a specific spatial frequency. Because of this, it can reduce the information not visible to the human eyes. Since, it concentrate just on still images. The redundancies existing, are of the following types,

i) Coding Redundancy: Present when less than optimal code words are used.
ii) Interpixel Redundancy: results from correlations between the pixels of an image.
iii) Psychovisual Redundancy: It is due to data that is ignored by the human visual system (i.e. visually non essential information).

## 1.3 Color Specification

The Y, Cb, and Cr components of one color image are characterized in YUV color coordinate, where Y is commonly called the luminance and Cb, Cr are ordinarily called the chrominance. The importance of luminance and chrominance is described as follows:

i) Luminance: got brightness of the light, which is corresponding to the aggregate vitality in the visible band.

ii) Chrominance: depict the apparent color tone of a light, which relies on upon the wavelength structure of light chrominance is turn described by two properties – hue and saturation.

a) Hue: Detail the color tone, which relies on upon the peak wavelength of the light
b) Saturation: Depict how unadulterated the color is, which relies on upon the spread or bandwidth

**1.4 FDCT and IDCT**

At the input to the encoder, source image samples are assembled into 8x8 blocks, moved from unsigned integers to signed integers, and input to the Forward Discrete Cosine Transform (FDCT). At the output from the decoder, the Inverse DCT (IDCT) yields 8x8 sample blocks to structure the reconstructed picture. The accompanying comparison Eq.1.1 and Eq.1.2 are the glorified numerical meanings of the 8x8 FDCT and 8x8 IDCT respectively.

$$F(u,v) = \alpha(u)\beta(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\left|\frac{u\pi(2x+1)}{2M}\right| \cos\left|\frac{v\pi(2y+1)}{2N}\right| \tag{1.1}$$

where,

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{M}}, & if\ u = 0 \\ \sqrt{\frac{2}{M}}, & otherwise \end{cases}$$

For most natural images, a significant number of the coefficients have small magnitudes and can be coarsely quantized (or discarded entirely) with little image distortion.

$$F(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \alpha(u)\beta(v) f(u,v) \cos\left|\frac{u\pi(2x+1)}{2M}\right| \cos\left|\frac{v\pi(2y+1)}{2N}\right| \tag{1.2}$$

It is an exceptional instance of the Fourier transform which eliminates the Sine components. Here, a 2D version of 1D variant evaluated twice is being utilized. It is worked two dimensionally considering 8 X 8 blocks of pixels. The ensuing information set is a block of frequency space components, the coefficients scaling the arrangement cosine terms, known as basis functions. The Principal component at column 0 and section 0, is known as the DC term, average frequency value of the entire block. The other 63 stipulations are AC segments, which articulate to the spatial frequencies that structures the information pixel block, by scaling the cosine terms inside the arrangement.

Since it is expected to manage pixel qualities extending from -128 to 127, each pixel is taken up by subtracting 128 from them. It involves 64 DCT coefficients, $C_{ij}$, where j and i stretches out from 0-7. The upper left coefficient, $C_{00}$, associates to the low frequencies of the original image block. As it move far from $C_{00}$ in every directions, $C_{77}$ relates to the highest frequency. The human eye is sensitive to low frequencies, reflected in quantization steps.

**1.5 Quantization**

The 8x8 blocks are ready for quantization. The coefficients $F_{u,v,}$ are real numbers, which will be stored as integers, hence it is necessary to round them off. It is done in a manner that encourages more prominent compression. Instead of essentially rounding the coefficients, $F_{u,v}$ is first partition by a quantizing factor and after that record round($F_{u,v}$ / $Q_{u,v}$)

An eminent and extremely important eccentricity of the standard is that in this step, quantization matrices contrasts the level of image compression and quality. This empowers the user to settle on quality levels running from 1 to 100, where 1 gives the poorest image quality and highest compression, while 100 gives the best quality and lowest compression. Therefore, the quality/ compression degree can be custom-made to suit distinctive needs.

A larger quantization factor is decided on the fact that human eye is not sensitive to high frequencies. Here q is responsible for the amount of compression and quality. This parameter, is an integer from 1 to 100. From q, Eq. 1.3 is used to calculate $\alpha$:

$$\alpha = \begin{cases} \frac{50}{q} \; if \; 1 \le q \le 50 \\ 2 - \frac{q}{50} \; if \; 50 \le q \le 100 \end{cases} \tag{1.3}$$

Notice in the above graph (Fig. 1.3), that higher values of $q$ give lower values of $\alpha$. The weights are then rounded off as

$$\text{round}(F_{u,v} / \alpha Q_{u,v}) \tag{1.4}$$

The above Eq. 1.4, results in information loss. The distinctive(default) coefficients recommended by the standard are mentioned below. $Q_l$ (Fig. 1.4) and $Q_c$ (Fig. 1.5), are the luminance and chrominance coefficients respectively:

$$Q_l = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Fig 1.3: Default Luminance Matrix

$$Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

Fig 1.4: Default Chrominance Matrix

**1.6 Encoder**

Entropy coding includes arranging the image components in a "zigzag" order utilizing run-length encoding (RLE) algorithm that groups comparable frequencies as one, and afterward utilizing Huffman coding. The value in the upper left corner represents the average of the entire block. Moving to the right builds the horizontal frequency while moving down increases the vertical frequency. The zeros can be ignored afterwards.

### 1.7 Comparison of Encoding Techniques

There are various lossless data compression encoding techniques. Some of main techniques are Huffman Coding, Run Length Encoding, Arithmetic Encoding.

| **Characteristics/Techniques** | **Huffman Coding** | **Run-Length Coding** | **Arithmetic Coding** |
|---|---|---|---|
| **Task** | Replaces fixed length codes by variable length codes. | Replaces runs of two or more characters with a single character or Code. | Uses occurrence and cumulative probabilities to represent the source message. |
| **Compression Ratio** | Average | Good | Very Good |
| **Compression Speed** | Very Fast | Fast | Slow |
| **Complexity** | Simple | Very Simple | Complex |
| **Decompression Speed** | Fast | Fast | Slow |

Table 1.1: Comparison of Encoding Techniques

From the above table, its obvious that Arithmetic Coding Technique is the most powerful encoding technique in terms of compression ratio. But, Huffman Coding and Run-Length Coding is still preferred over Arithmetic Coding because of the compression and decompression speed and complexity.

### 1.8 Image Inpainting/Restoration Techniques

The dominant type of redundancy within images originates from their representation method. Each digital image is composed of discrete points, called pixels. The value relevant to each pixel is the result of sampling from light or color intensity in the original image domain. Natural images consist of separate areas indicating the object surfaces or sceneries. Because the light intensity and color in such areas are approximately constant, the relevant values for pixels are highly correlated. Every pixel in such areas is likely to be of the same or very close value compared with the adjacent pixels. In this case, images suffer from a high level of spatial correlation. Hence, representing the image by storing all pixel values results in a large amount of redundancy.

Instead of using a simple copy, it can approximate the indefinite pixels of the area by a linear combination of several best matching patches (i.e. of K nearest neighbors, K-NN), this way exploiting self-similarities within the image. The authors utilized the weights of the linear combination to compute a similarity kernel. This methodology is motivated from texture synthesis [8] and the non-local means (NLM) algorithm, used for de-noising.

Instance Based Learning algorithms basically stores some or the majority of the training examples and defers any speculation exertion, until another query instance is further anticipated. Actually known cases and experiences is applied for particular cases or experiences to new circumstances. They can along these lines, construct query specific local models, which endeavor to match the training samples just in an area around the query instance alluded to as lazy learning method. It simply discover a set of the nearest neighbors and develops a local model focused on them. Nearest Neighbor (NN), K Nearest Neighbor (KNN) and Locally Weighted Regression (LWR). NN local models just choose the closest point and utilize its output value. The outputs of the nearby points are averaged and utilized by KNN.

### 1.8.1 Image Restoration with K>3

The output is a class membership in KNN Classification. Characterization of an entity is done by a majority vote of its neighbors. The object which is most normal among its K closest neighbors is relegated (k is a positive number, typically small). The object is simply assigned to the class of that single closest neighbor when $k = 1$.

Here, K>3 is considered, i.e. an aggregate of k neighbors of every pixel can be considered in a closed window of 8x8. In the 2d matrix of image components, every component has a certain correlation with its closest components. We can write algorithms by considering this property, and use the mean of all the closest neighbors to replace a damaged pixel. Hence a decent level of restoration is guaranteed by this method. The algorithm proposed will complete an iterative methodology wherein the mean intensity is discovered and further substitution of noisy pixel is carried out. Consider an input picture $P_i$, given us a chance to characterize a pixel at a position (q,r) in the info picture. The computation of the likelihood of event of each one neighbor of $P_i(q,r)$ is performed first. The mean worth is acquired by utilizing the accompanying representation Eq. 1.5, for a sum of k neighbors in that window.

$$M = \sum_{i=1}^{K} X_i P(X_i) \tag{1.5}$$

Eq. 1.5 gives the mean of all neighboring points of a specific pixel. This gives a value what we call as a "good pixel value". Hence the good pixel value replaces the central corrupt pixel. This guarantees the removal of the tainted pixels and restoration of the given image.

# CHAPTER 2
# LITERATURE REVIEW

**Qiang LI et.al [1]** proposed a novel algorithm that uses compressed sensing (CS) in the frequency domain rather than most existing algorithms which are pixel based, to recreate corrupted images. With a specific end goal to reconstruct image, the authors first disintegrated the picture into two functions with diverse basic characteristics - structure component and textual component. The method ensured good image quality with the structure being decently restored, and the PSNR is generally high.

**Christine Guillemot et.al [2]** depicted an examplar-based picture inpainting algorithm locally linear neighbor embedding technique with low-dimensional neighborhood representation (LLE-LDNR). The inpainting algorithm first searches the K nearest neighbors (K-NN) of the input patch to be filled-in and linearly combine them with LLE-LDNR to synthesize the missing pixels. Linear regression is then introduced for enhancing the K-NN search. The results likewise demonstrated further gains when utilizing the proposed improved K-NN search utilizing linear subspace mappings in the context of inpainting.

**Li Zhiqiang et.al [3]** disintegrated the coding algorithm of JPEG, advances the JPEG encoder and decoder control processes. They showed the encryption and decryption of image by using Logistic sequence. Results demonstrated that this method not just save storage space for other helpful data, additionally enhanced the transmission efficiency and security in the transmission process. The article combines JPEG compression algorithm with chaotic encryption algorithm, which can viably save the storage space for image and guarantees the secure transmission of image information.

**Vahid Bastani et.al [4]** proposed an algorithm for image compression focused around an image inpainting system. Initially the image regions that can be precisely recuperated are located. At that point, to lessen the information, data of such locales is evacuated. The remaining information other than essential details for recovering the removed regions are encoded to deliver output data. The PSNR and SSIM of the sample images, demonstrated that the proposed technique outperformed JPEG at high compression ratios, for example, 1:40 (0.2 bpp) and were more outstanding in low structured and low textured pictures.

**En-hui Yang et.al [5]** proposed a novel algorithm to find the ideal SDQ coefficient files as run-size sets among all conceivable candidates given that the other two parameters are fixed. Taking into account this algorithm, they formed an iterative algorithm to mutually optimize the run-length coding, Huffman coding and quantization step sizes. The proposed iterative algorithm attains a compression performance better than any formerly known JPEG compression results and even surpasses the cited PSNR results of some state-of-the-art wavelet-based image coders like Shapiro's embedded zerotree wavelet algorithm at the regular bit rates under comparison.

**Suyash P. Awate et.al [6]** proposed a paper that depicts a novel unsupervised, information theoretic, adaptive filter (UINTA), that enhances the consistency of pixel intensities from their neighborhoods by diminishing their joint entropy. Thus, UINTA naturally finds the statistical properties of the signal and can in this way restore a wide range of images. The paper depicts the plan to minimize the joint entropy measure and presents several important practical contemplations in evaluating neighborhood statistics. It exhibits an arrangement of results on both real and synthetic.

**Timothy K. Shih et.al [7]** proposed a versatile system, which is focused around a color interpolation mechanism on Digital Image Inpainting. The repairing technique checks the encompassing data of a damaged pixel and chooses the scope of references that can be utilized to process an interpolated color. The framework was tried on more than 2000 images incorporating painting, photograph, and cartoon drawing. The assessment demonstrated that their mechanism produces a very good result.

**Gopal Lakhani et.al [8]** introduced a minor change to the Huffman coding of the JPEG baseline compression algorithm to exploit this redundancy. It is a well observed characteristic that when a DCT block is traversed in the zigzag order, the AC coefficients generally decrease in size and the run-length of zero coefficients increase in number. For this reason, DCT blocks are partitioned into groups with the goal that each one band can be coded utilizing a different code table. Three implementations are introduced, which all move the end-of- block marker up amidst DCT block and use it to show the band limits. Experimental results are compare reduction in the code size acquired by their techniques with the JPEG sequential-mode Huffman coding and arithmetic coding methods.

**Mitchell A. Golner et.al [9]** proposed a region based variable quantization scheme, where the quantization granularity in diverse preselected regions of the image is varied at the discretion of the user. The techniques developed in this work are compatible with the popular JPEG Still Image Standard for compression of continuous-tone grey-scale and color images. Further, region selection techniques and algorithms that complement variable quantization techniques are introduced. The paper introduced three masks: step, linear interpolated, and raised cosine interpolated, that control the transition in the quantization granularity between regions of diverse compression ratios in an image. The paper likewise incorporates a point by point discussion of simulation results utilizing the proposed methodology.

**Q. Huynh-Thu et.al [10]** Experimental data introduced in the paper clearly exhibited the scope of application of peak signal-to-noise ratio(PSNR) as a video quality metric. Its shown that PSNR is a legitimate quality measure, unless the video content and codec type are not changed. However,when the substance is changed, connection between subjective quality and PSNR is highly reduced. Hence, PSNR can't be a reliable method for assessing the video quality across different video contents.

**Zhou Wang et.al [11]** reviewed on the reasons why we ought to utilize or leave the respected (however maybe hoary) MSE. The author likewise gave an emerging alternative for signal fidelity measures and discussed about their potential application to a wide variety of problems. The message the authors attempted to send is, not that one ought to forsake utilization of the

MSE nor to indiscriminately change to some other specific signal fidelity measure. Rather, to make the point that there are capable, simple to-utilize, and straightforward choices that may be conveyed relying upon the application environment and needs. It is normal that the MSE will keep on being broadly utilized as a sign fidelity measure

**M. Bertalmio et.al [12]** introduced a novel algorithm for automatic digital inpainting, being its main inspiration to replicate the essential methods utilized by expert restorators. The only user interaction required by the algorithm here introduced is to mark the regions to be inpainted. Additionally, since the inpainting algorithm here displayed can be utilized to restore damaged photos as well as to remove undesired objects and writings on the picture, the regions to be inpainted must be marked by the client.

**G. Guleryuz et.al [13]** created a system that are designed for the recovery of missing regions utilizing the minimum mean-squared error (mse) rule and an implicit statistical model. While the primary applications was the estimation of missing locales in pictures and video, its reasonable that the created strategies can be summed up to handle missing "areas" in different sorts of signals and they can likewise be summed up to suit different applications. For instance, the procedures can be utilized as a major aspect of an encoder that reduces redundancy by predictively encoding signals, images, and video, or conveyed in applications that require prediction in more general situations.

**G. Guleryuz et.al [14]** gave a broad arrangement of simulation examples that show the performance of the derived algorithm. Through those examples the author discussed about the properties needed of good transforms that will bring about successful estimates [in the sense of mean-squared error (mse)] over various region types. We further demonstrate the performance and robustness of remarkable changes in estimating a broad set of image regions.

**Alexander Wong et.al [15]** introduced a novel methodology with the problem of image inpainting through the utilization of nonlocal-means.In the proposed algorithm, the author utilized nonlocal image information from numerous samples within the image. The contribution of each specimen to the reconstruction of a target pixel is resolved utilizing a weighted similarity function and aggregated to shape the missing information. Exploratory results demonstrate that the proposed technique yields quantitative and subjective changes contrasted with the current exemplar-based approach. The proposed methodology can likewise be coordinated into existing exemplar-based inpainting procedures to provide improved visual quality.

**Alexandru Telea et.al [16]** proposed a new inpainting algorithm taking into account propagating an image smoothness estimator along the image gradient, similar to [32]. The author evaluated the picture smoothness as a weighted average over a known image neighborhood of the pixel to inpaint. He regarded the missing regions as level sets and utilized the fast marching method (FMM) to propagate the image information.

**David Tschumperl´e et.al [17]** introduced another tensor-determined PDE, regularizing images while considering the curves of particular fundamental integral curves. The paper demonstrated that this limitation is especially appropriate for the protection of thin structures in an image restoration process. A direct connection is made between the proposed equation and a constant definition of the LIC's (Line Integral Convolutions).

**N. Ahmed et.al [18]** contrasted the performance of the DCT and KLT, DFT, and the identity transforms, utilizing the rate-distortion criterion. This performance basis gives a measure of the data rate R that can be attained to while still maintaining a fixed distortion D, for encoding purposes.

**Leung, A. E et.al [19]** The surface blend procedure develops another picture outward from an introductory seed, one pixel at once. A Markov irregular field model is assumed, and the contingent appropriation of a pixel given every one of its neighbors incorporated so far is evaluated by querying the sample image and discovering all comparable neighborhoods. The level of irregularity is controlled by a solitary perceptually instinctive parameter.

**A. Criminisi et.al [20]** algorithm is proposed for removing substantial objects from digital images. The test was to fill in the gap that is deserted in an outwardly conceivable manner. A best-first algorithm was proposed in which the confidence in the blended pixel qualities is spread in a way like the proliferation of data in inpainting. The actual color values are registered utilizing model based combination.

**C. Barnes et.al [21]** Synthesized complex texture and image structures that resembles input imagery by image retargetting, image completion and image reshuffling.

**Y. Wexler et.al [22]** present a technique for space-time consummation of substantial space-time "holes" in video sequences of complex element scenes. The missing bits are filled-in by examining spatio-temporal patches from the accessible parts of the feature, while implementing worldwide spatio-temporal consistency between all patches in and around the opening.

**Sun Z. X et.al [23]** presents a novel examplar-based inpainting algorithm through investigating the sparsity of natural image patches. Two novel ideas of sparsity at the patch level are proposed for displaying the patch priority and patch representation, which are two vital steps for patch proliferation in the examplar-based inpainting methodology.

# CHAPTER 3
# PRESENT WORK

---

**3.1 Problem formulation**

Existing strategies can be ordered into the accompanying classifications:

i) The principal class concerns diffusion-based methodologies which proliferate level lines (called isophotes) utilizing partial differential equations (PDE) or variational systems.

ii) The second classification concerns examplar-based inpainting methods which have been motivated from texture synthesis procedures. These methods exploit image statistical and equivalence toward oneself priors.

iii) An alternate class of methodologies concerns techniques utilizing sparsity priors.

iv) Most existing algorithms are pixel based, which develop a statistical model from image characteristics. One of the primary burdens of these methodologies is that, their viability is constrained by the surrounding pixels of the destroyed part.

Images in the frequency domain contain sufficient data for image inpainting and can be utilized as a part of data recreation. High frequency indicates image edges or textures, which motivates conducting image inpainting in the frequency domain.

A novel KNN method is proposed which will utilize the DCT coefficients in the frequency domain to remove the distortions. We look for an adequate representation for the functions and utilize the DCT coefficients of this representation to produce an over-complete dictionary. This strategy can be implemented during compression with JPEG Algorithm, after the the DCT step to get a more optimized result.

### 3.2 Objectives

The two main objective of this paper is inpainting and compressing images with noise(after denoising/inpainting). As discussed in the introduction section, it's preferable that the image is first removed from any distortions before applying compression to get a more optimized output.

1) To improve the compression ratio of the distorted images, as compared to the baseline JPEG Algorithm with images having minor distortions and cracks.
2) To remove the distortions/redundancy from an image before applying compression (improvement in compression ratio).
3) To apply KNN algorithm on the frequency domain and use the DCT Coefficients to remove the redundancy.
4) To implement the proposed system in MATLAB and obtain the results.

### 3.3 Methodology

The methodology adopted, consists of the following steps:
1) **Exploration**: This approach is used to collect information about the techniques mentioned in the papers from the journals.
2) **Reading**: This step is for gaining a thorough knowledge about the techniques by continuous reading.
3) **Deduction**: Summing up the main steps/concepts, according to the field of study.
4) **Conclusion**: Getting into a particular conclusion from the ideas gained from the above steps. The steps are repeated untill the conclusion of the proposed approach is finalized.

#### 3.3.1 Proposed Approach

1) Obtain image (having cracks and minor distortions).
2) Divide it into 8x8 matrix components.
3) Converts to Grayscale.
4) Convert to Black and White (logical class) and save the positions of 1(luminance) from the matrix.
5) Apply DCT to the matrix obtained from step 3.

6) Apply KNN Algorithm to the positions obtained from step 4, in the DCT Coefficients.

7) Apply Quantization to compress the data.

8) Apply Variable Length Coding which includes Zigzag filtering and Run-Length Encoding or Huffman Encoding.

### 3.3.2 Work Methodology

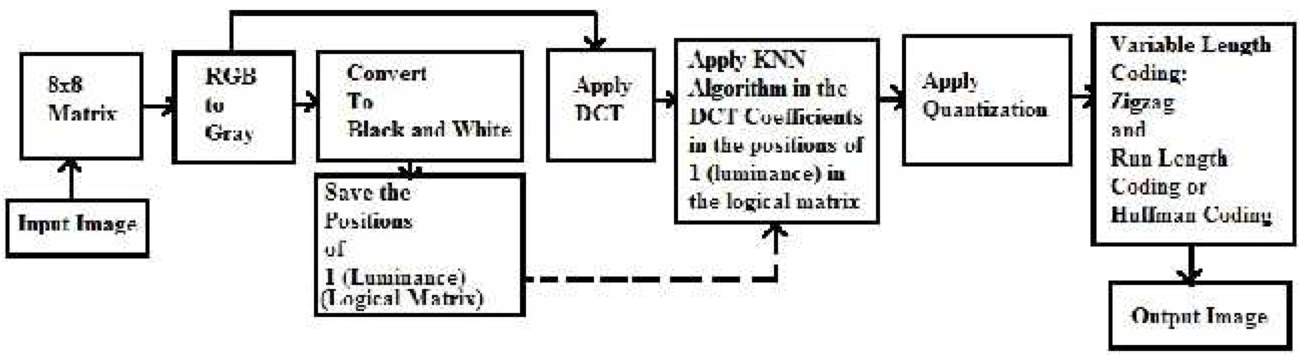



Fig 3.1: Methodology

The image is converted into Black and White to retrieve the positions of cracks/distortions. In this, only luminance is shown with 1(the cracks and distortions) in the matrix. Rest every chroma component is 0. This is known as logical class.

To do restoration, the closest neighbors of a pixel is considered. Here, K>3 is considered, i.e. an aggregate of k neighbors of every pixel can be considered in a closed window of 8x8 of the DCT Coefficients. In the 2d matrix of image components, every component has a certain correlation with its closest components. The algorithm is written by considering this property, and using the mean of all the closest neighbors to replace a damaged pixel. Hence a decent level of restoration is obtained by this method. The algorithm proposed completed an iterative methodology wherein the mean intensity is discovered and further substitution of noisy pixel is carried out.

# CHAPTER 4
# RESULTS AND DISCUSSIONS

---

### 4.1 Image Quality Evaluation

The viability of the algorithm is tested by using three frameworks. First is the Mean Squared Error (MSE) of the image with also its Peak Signal to Noise Ratio (PSNR), which is the second quality index. The results are shown in tables (4.1,4.3,4.5,4.7,4.9) Third, the comparison of the compression ratio between an image with cracks and its inpainted output is shown. The results are shown in tables (4.2,4.4,4.6,4.8)

#### 4.1.1 MSE and PSNR

A conventional image quality index is the peak signal-to-noise ratio (PSNR) [10], which is the ratio of the squared image intensity dynamic range to the mean squared difference of the original and distorted images, Eq. 4.1.1. It is widely used for the estimation of quality in lossy image compression algorithms. This index is popular for its simplicity; however, it loses its advantages compared with natural human perception [11]. Hence, the subjective judgment of the viewer additionally is viewed as an essential measure, maybe, being the most critical measure for judging an image.

$$PSNR = 10 \times \log_{10}\left(\frac{(2^n-1)^2}{MSE}\right) \tag{4.1.1}$$

where,

$$MSE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}[l(i,j) - k(i,j)]^2 \tag{4.1.2}$$

### 4.1.2 Compression Ratio

The compression ratio of an image is given by comparing the original size of the given image with its compressed size. The formula is shown in Eq 4.1.3

$$\text{Compression Ratio} = \frac{\text{CompressedSize}}{\text{Originalsize}} \tag{4.1.3}$$

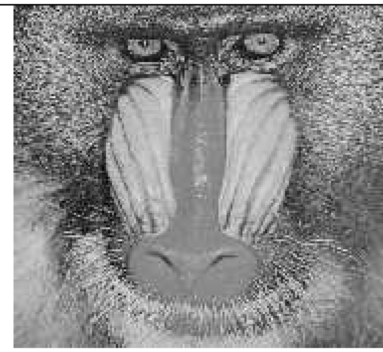
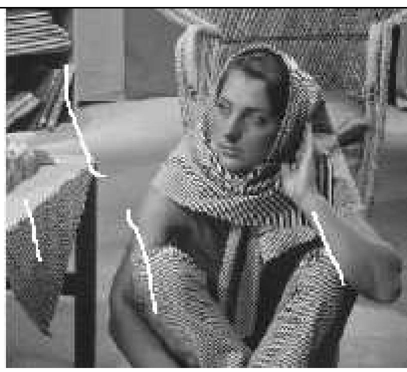
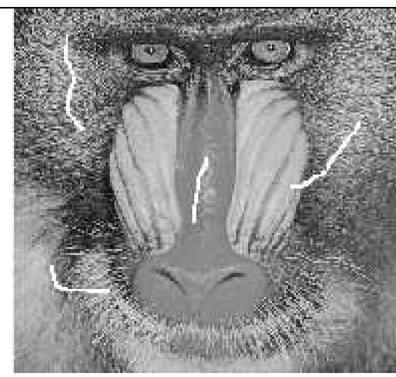
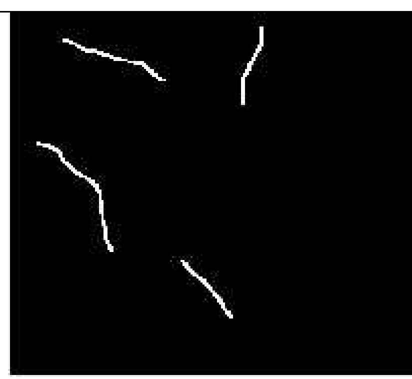
## 4.2 Experiment Results

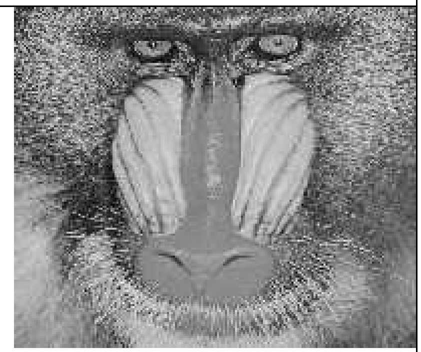
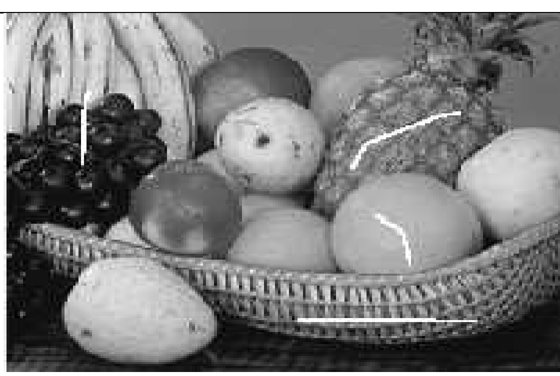
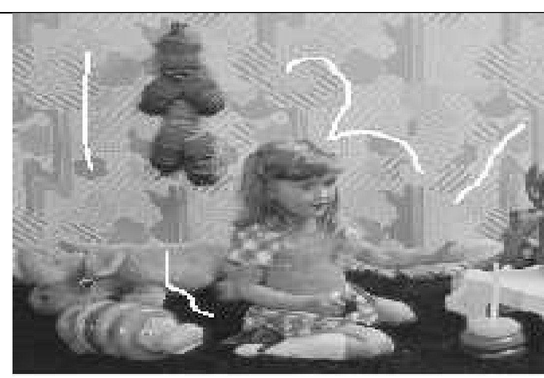
The proposed algorithm is implemented in Matlab 2013a version. The results obtained contains the output both after inpainting and inpainting with compression. The proposed approach was implemented on the real world datasets such as the IVC Database [31], Standard Images [32], Online Database by Computer Vision [33] and the tid2008 Database [34], used in Digital Image Processing. The results obtained are on table 4.1 to 4.9.

### 4.2.1 Experiments with grayscale images

The initial datasets were of grayscale and was superimposed with user defined mask randomly. The mask were of white background. The results obtained after applying the proposed approach were impressive and those mask were efficiently restored with the original pixels.

Before the application of the K-Nearest Neighbor algorithm to the saved positions of the white background, the algorithm first checks if the image is of proper size, possibly divisible by 8. If not, it adds rows and columns to the image such that it becomes a multiple of 8, so that the calculation of DCT Coefficients is properly done without any errors.

| .ORIGINAL IMAGES | | |
|---|---|---|
|  |  |  |
| **BARBARA (512x512)** | **CLOWN (512x512)** | **MANDARIN 512x512** |
| **USER APPLIED MASK** | | |
|  |  |  |
| **BARBARA** | **CLOWN** | **MANDARIN** |
| **MASK DETECTION** | | |
|  |  |  |
| **BARBARA** | **CLOWN** | **MANDARIN** |

<table>
<tr><th colspan="6">RESTORED IMAGES</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">BARBARA</td><td colspan="2">CLOWN</td><td colspan="2">MANDARIN</td></tr>
<tr><th colspan="6">ORIGINAL IMAGES</th></tr>
<tr><td colspan="3"></td><td colspan="3"></td></tr>
<tr><td colspan="3">FRUIT (512x512)</td><td colspan="3">ISABEL (512x512)</td></tr>
<tr><td colspan="6"></td></tr>
<tr><th colspan="6">USER APPLIED MASK</th></tr>
<tr><td colspan="3"></td><td colspan="3"></td></tr>
<tr><td colspan="3">FRUIT</td><td colspan="3">ISABEL</td></tr>
<tr><td colspan="6"></td></tr>
</table>

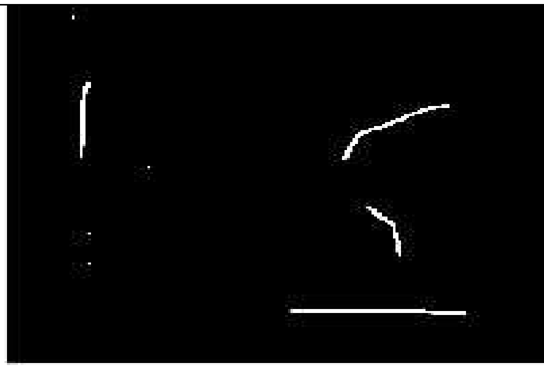
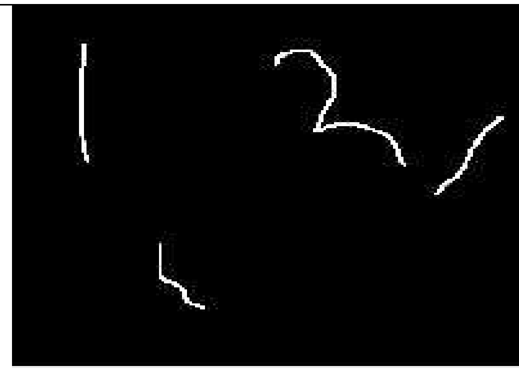
| MASK DETECTION | |
|---|---|
|  |  |
| **FRUIT** | **ISABEL** |
| | |
| **RESTORED IMAGES** | |
|  |  |
| **FRUIT** | **ISABEL** |

Fig 4.1: Images from the IVC Database of size 512x512.

| **Images** | **MSE** | **PSNR** |
|---|---|---|
| barbara_gray | 6.49 | 40.005 |
| clown_gray | 3.68 | 42.50 |
| fruit_gray | 7.87 | 39.17 |
| isabel_gray | 6.17 | 40.22 |
| mandrin_gray | 7.88 | 39.16 |

Table 4.1: MSE and PSNR Values of the images from IVC Database.

| **BARBARA 512x512** | |
|---|---|
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |
| **CLOWN (512x512)** | |
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |
| **FRUIT (512x512)** | |
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |
| | |

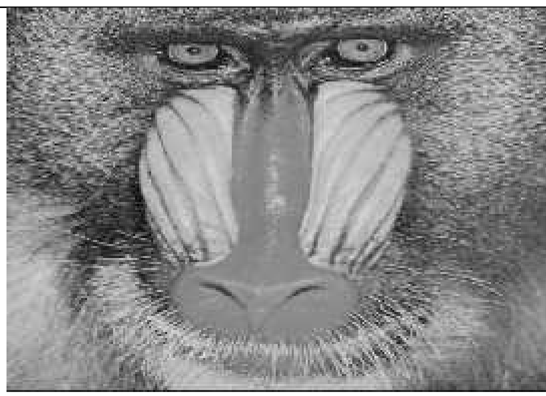
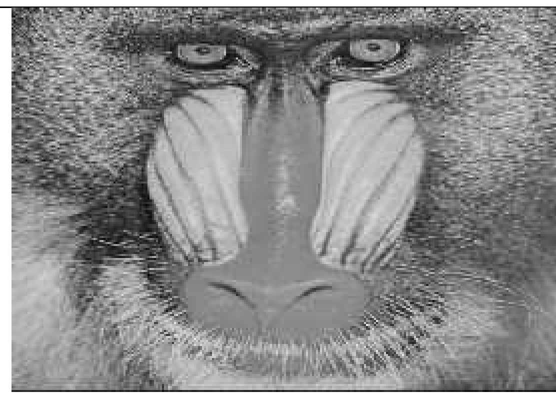
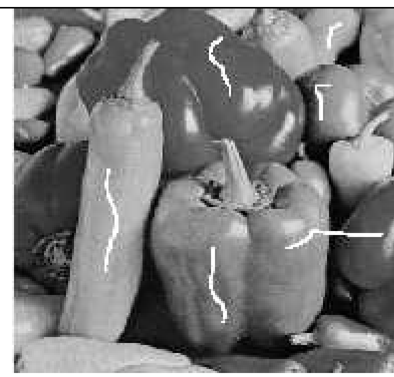
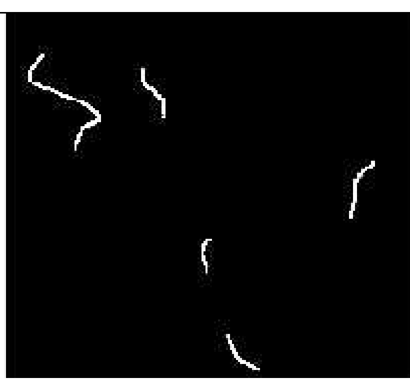
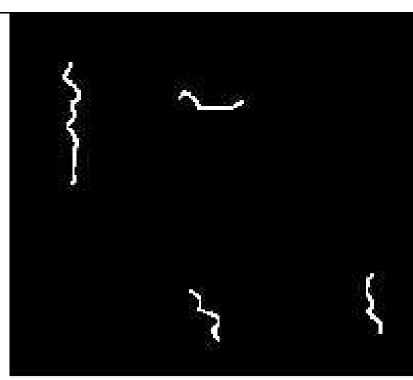
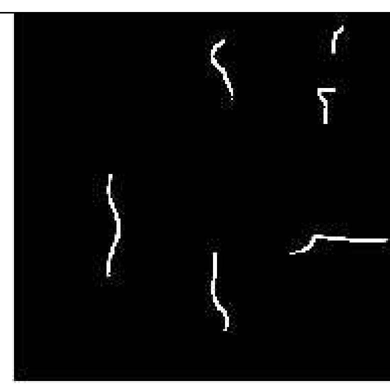
| **ISABEL 512x512** | |
|---|---|
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |
| **MANDARIN 512x512** | |
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |

Fig 4.2: Original Image and its Compressed output of the images from the IVC Database.

| **COMPRESSION RATIO** | | |
|---|---|---|
| **Images** | **Before Inpainting** | **After Inpainting** |
| barbara_gray (Original Size:768 Kb) | 24.38 (Size: 31.5 Kb) | 25.18 (Size: 30.5 Kb) |
| clown_gray (Original Size:768 Kb) | 28.23 (Size:27.2 Kb) | 29.88 (Size:25.7 Kb) |
| fruit_gray (Original Size:768 Kb) | 28.76 (Size:26.7 Kb) | 29.65 (Size:25.9 Kb) |
| isabe_gray (Original Size:768 Kb) | 28.98 (Size:26.5 Kb) | 30.23 (Size:25.4 Kb) |
| mandr_gray (Original Size:768 Kb) | 17.29 (Size:44.4 Kb) | 17.53 (Size:43.8 Kb) |

Table 4.2: Compression Ratio and the Original and Compressed Size of the images from IVC Database before and after Inpainting.

| **ORIGINAL IMAGES** | | |
|---|---|---|
|  |  |  |
| **LENA (512x512)** | **LAKE (512x512)** | **PEPPERS (512x512)** |
| | | |
| **USER APPLIED MASK** | | |
|  |  |  |
| **LENA** | **LAKE** | **PEPPERS** |
| | | |
| **MASK DETECTION** | | |
|  |  |  |
| **LENA** | **LAKE** | **PEPPERS** |

| RESTORED IMAGES | | |
|---|---|---|
|  |  |  |
| **LENA** | **LAKE** | **PEPPERS** |

Fig 4.3: Images from the Standard Image Database of size 512x512.

| **Images** | **MSE** | **PSNR** |
|---|---|---|
| lake_gray | 1.87 | 45.40 |
| lena_gray | 2.55 | 44.05 |
| Peppers_gray | 2.06 | 44.98 |

Table 4.3: MSE and PSNR Values of the images from the Standard Image Database.

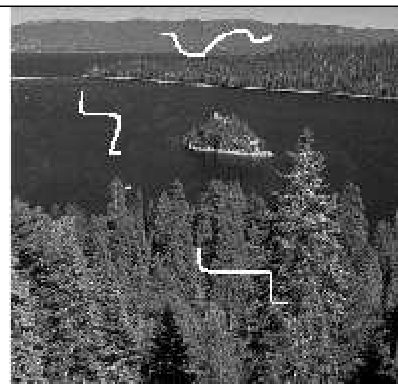
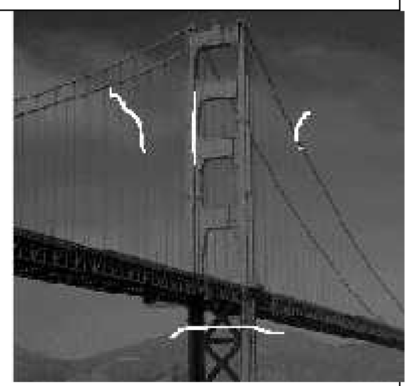
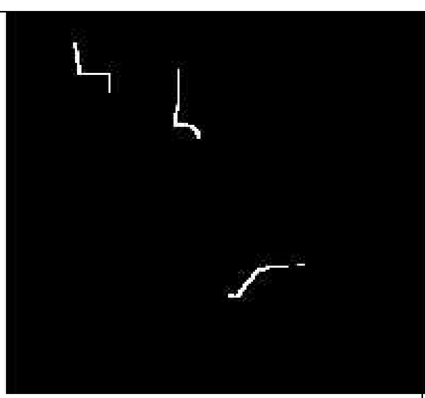
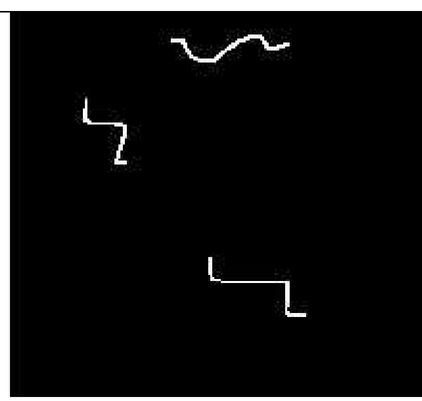
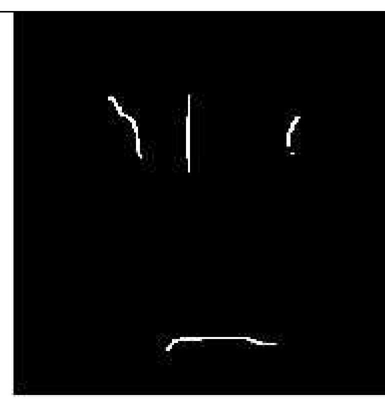
| LENA 512x512 | |
|---|---|
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |

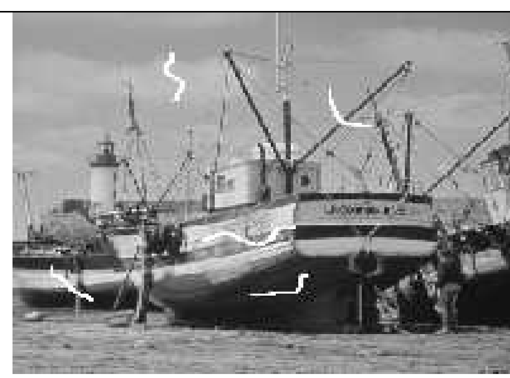
| **LAKE (512x512)** | |
|---|---|
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |
| **PEPPERS 512x512** | |
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |

Fig 4.4: Original Image and its Compressed output from the Standard Database.

| **COMPRESSION RATIO** | | |
|---|---|---|
| **Images** | **Before Inpainting** | **After Inpainting** |
| lake_gray (Original Size:768 Kb) | 25.01 (Size:30.7 Kb) | 26.03 (Size:29.5 Kb) |
| lena_gray (Original Size:768 Kb) | 29.88 (Size:25.7 Kb) | 31.60 (Size:24.3 Kb) |
| Peppers_gray (Original Size:768 Kb) | 32.26 (Size:23.8 Kb) | 34.43 (Size:22.3 Kb) |

Table 4.4: Compression Ratio and the Original and Compressed Size of the images from Standard Image Database before and after Inpainting.

| ORIGINAL IMAGES | | |
|---|---|---|
|  |  |  |
| **10 (512x512)** | **15 (512x512)** | **25 (512x512)** |
| | | |
| **USER APPLIED MASK** | | |
|  |  |  |
| **10** | **15** | **25** |
| | | |
| **MASK DETECTION** | | |
|  |  |  |
| **10** | **15** | **25** |

<table>
<tr><th colspan="6">RESTORED IMAGES</th></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2">10</td><td colspan="2">15</td><td colspan="2">25</td></tr>
<tr><td colspan="6"></td></tr>
<tr><th colspan="6">ORIGINAL IMAGES</th></tr>
<tr><td colspan="3"></td><td colspan="3"></td></tr>
<tr><td colspan="3">34 (512x512)</td><td colspan="3">48 (512x512)</td></tr>
<tr><td colspan="6"></td></tr>
<tr><th colspan="6">USER APPLIED MASK</th></tr>
<tr><td colspan="3"></td><td colspan="3"></td></tr>
<tr><td colspan="3">34</td><td colspan="3">48</td></tr>
<tr><td colspan="6"></td></tr>
</table>

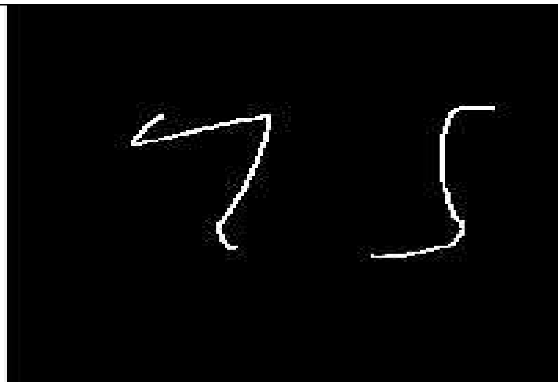
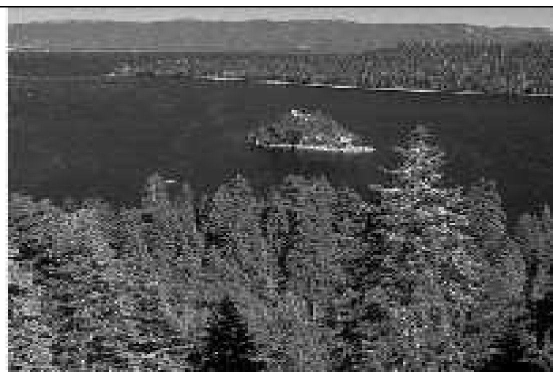
| **MASK DETECTION** | |
|---|---|
|  |  |
| **34** | **48** |
| | |
| **RESTORED IMAGES** | |
|  |  |
| **34** | **48** |

Fig 4.5: Few Images from the Online database by Computer Vision of Size 512x512

| **Image No** | **MSE** | **PSNR** |
|---|---|---|
| 10 | 2.93 | 43.45 |
| 15 | 3.79 | 42.33 |
| 25 | 2.28 | 44.54 |
| 34 | 12.38 | 37.20 |
| 48 | 3.35 | 42.88 |

Table 4.5: MSE and PSNR values of the images from the Online database by Computer Vision

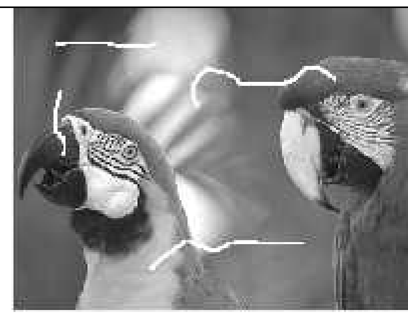
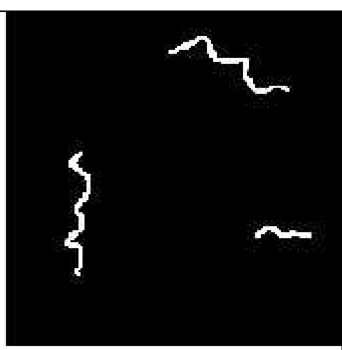
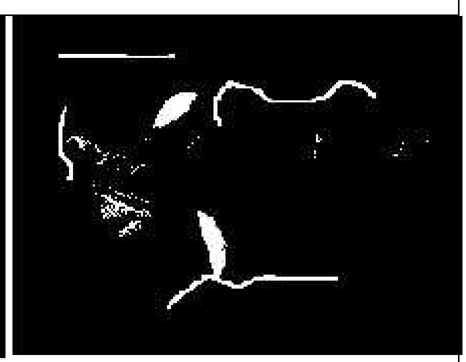
<table>
<tr><td colspan="2">10 (512x512)</td></tr>
<tr><td></td><td></td></tr>
<tr><td>ORIGINAL IMAGE</td><td>COMPRESSED IMAGE</td></tr>
<tr><td colspan="2">15 (512x512)</td></tr>
<tr><td></td><td></td></tr>
<tr><td>ORIGINAL IMAGE</td><td>COMPRESSED IMAGE</td></tr>
<tr><td colspan="2">25 (512x512)</td></tr>
<tr><td></td><td></td></tr>
<tr><td>ORIGINAL IMAGE</td><td>COMPRESSED IMAGE</td></tr>
</table>

| **34 (512x512)** | |
|---|---|
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |
| **48 (512x512)** | |
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |

Fig 4.6: Original and its Compressed output of the images from the Online database by Computer Vision

| **COMPRESSION RATIO** | | |
|---|---|---|
| **Images** | **Before Inpainting** | **After Inpainting** |
| 10 (Original Size:768 Kb) | 31.86 (Size:24.1 Kb) | 32.40 (Size:23.7 Kb) |
| 15 (Original Size:768 Kb) | 19.69 (Size: 39.0Kb) | 20.26 (Size:37.9 Kb) |
| 25 (Original Size:768 Kb) | 40.85 (Size:18.8 Kb) | 42.90 (Size:17.9 Kb) |
| 34 (Original Size:768 Kb) | 20.81 (Size:36.9 Kb) | 21.39 (Size:35.9 Kb) |
| 48(Original Size:768 Kb) | 28.98 (Size:26.5 Kb) | 30.00 (Size:25.6 Kb) |

Table 4.6: Compression Ratio and the Original and Compressed Size of the images from the Online database by Computer Vision

<table>
<tr><th colspan="3">ORIGINAL IMAGE</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>I04 (332x326)</td><td>I07 (512x384)</td><td>I23 (512x384)</td></tr>
<tr><th colspan="3">USER APPLIED MASK</th></tr>
<tr><td></td><td>
</td><td></td></tr>
<tr><td>I04 (332x326)</td><td>I07 (512x384)</td><td>I23 (512x384)</td></tr>
<tr><th colspan="3">MASK DETECTION</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>I04 (332x326)</td><td>I07 (512x384)</td><td>I23 (512x384)</td></tr>
</table>

| RESTORED IMAGES | | |
|---|---|---|
|  |  |  |
| **I04 (332x326)** | **I07 (512x384)** | **I23 (512x384)** |

Fig 4.7: Few Images from the tid2008 database of variable sizes

| **Images** | **MSE** | **PSNR** |
|---|---|---|
| I04 (332x326) | 1.25 | 47.14 |
| I07 (512x384) | 7.39 | 39.43 |
| I23 (512x384) | 31.48 | 33.14 |

Fig 4.7: MSE and PSNR values of the images from the tid2008 database

| I04 (332x326) | |
|---|---|
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |

| **I07 (512x384)** | |
|---|---|
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |
| | |
| **I23 (512x384)** | |
|  |  |
| **ORIGINAL IMAGE** | **COMPRESSED IMAGE** |

Fig 4.8: Original and its Compressed output of the images from the tid2008 database

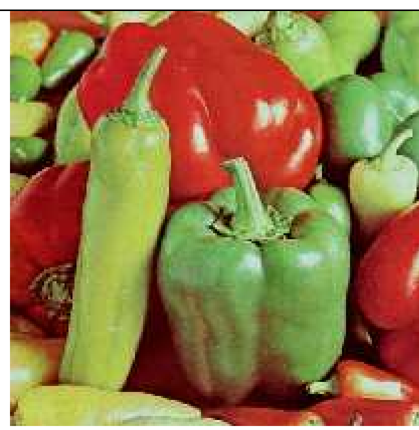
| **COMPRESSION RATIO** | | |
|---|---|---|
| **Images** | **Before Inpainting** | **After Inpainting** |
| I04 (332x326) (Original Size:317 Kb) | 31.38 (Size:10.1 Kb) | 35.06 (Size:9.04 Kb) |
| I07 (512x384) (Original Size:576 Kb) | 30.15 (Size:19.1 Kb) | 30.80 (Size:18.7 Kb) |
| I23 (512x384) (Original Size:576 Kb) | 38.14 (Size:15.1 Kb) | 42.04 (Size:13.7 Kb) |

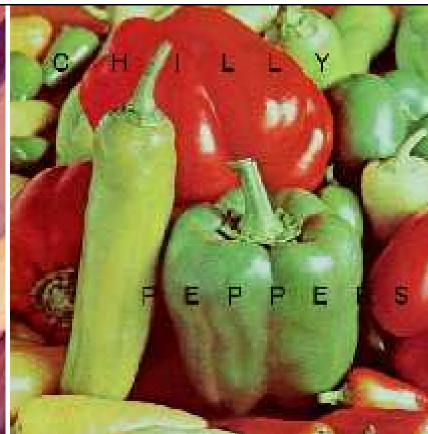
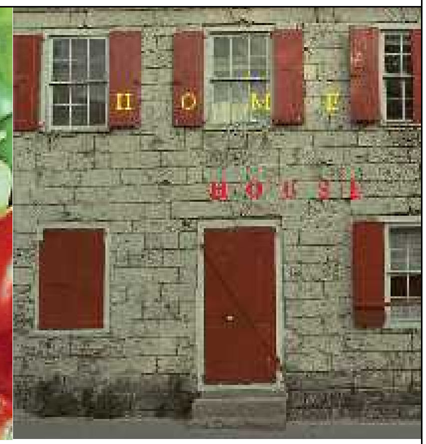
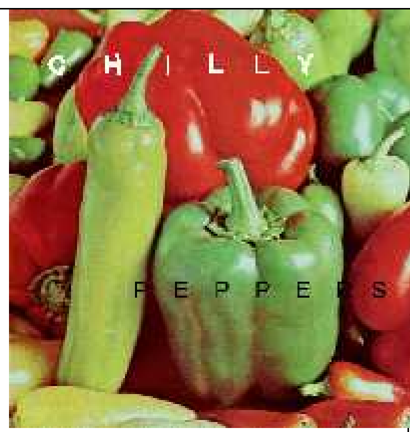
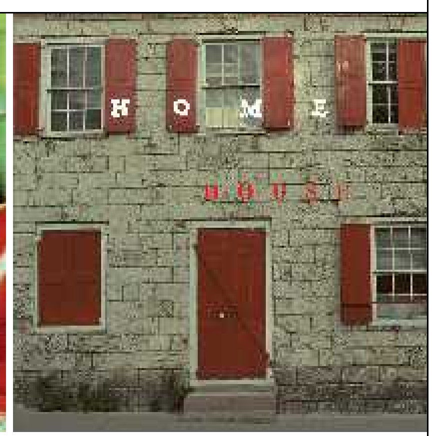
Table 4.8: Compression Ratio and the Original and Compressed Size of the images from the tid2008 database.

According to the above tables, it's clear that the algorithm gave a high PSNR value (>33), as high as 47.14. A high PSNR Value shows that the algorithm gave an optimum result with the cracks being well restored. The inpainted and compressed images are visually attractive with less distortions. Also the Compression Ratio is always found to be better and optimum after inpainting is applied and their original pixel is restored first.

### 4.2.2 Experiments with Colored Images (RGB Images)

The first experiments were on gray scale images with multiple datasets. The next example (Fig:4.9) are of color images. But the experiment is restricted to the inpainting module. The Compression algorithm didn't work with RGB images. With a colored image having the three color channels, the inpainting algorithm first split the image into R, G and B in the DCT Coefficients, and process them individually using the technique, lastly merging them together. So the strategy similarly contributes in inpainting color images, and the primary process are the same to grayscale images. It can be seen that the visual performance is better, because of data remuneration from the three channels.

| ORIGINAL IMAGES | | |
|---|---|---|
|  |  |  |
| **LENA (512x512)** | **PEPPERS (512x512)** | **HOUSE (512x512)** |

<table>
<tr><td colspan="3">USER WRITTEN TEXT</td></tr>
<tr>
<td>
</td>
<td>
</td>
<td></td>
</tr>
<tr><td>LENA (512x512)</td><td>PEPPERS (512x512)</td><td>HOUSE (512x512)</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3">USER APPIED MASK</td></tr>
<tr>
<td>
</td>
<td></td>
<td></td>
</tr>
<tr><td>LENA (512x512)</td><td>PEPPERS (512x512)</td><td>HOUSE (512x512)</td></tr>
</table>

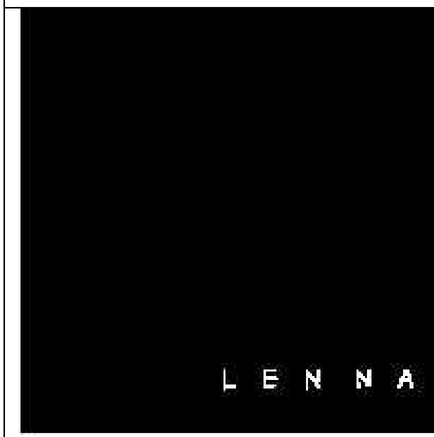
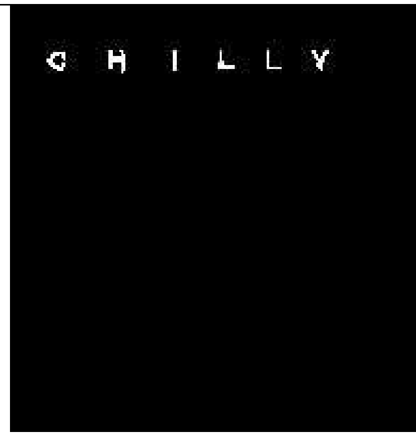
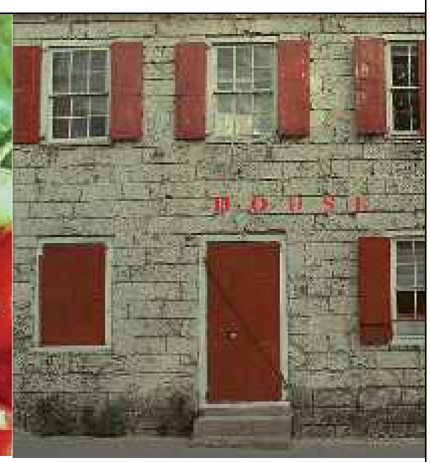
| MASK DETECTION | | |
|---|---|---|
|  |  |  |
| **LENA (512x512)** | **PEPPERS (512x512)** | **HOUSE (512x512)** |
| | | |
| **RESTORED IMAGES** | | |
|  |  |  |
| **LENA (512x512)** | **PEPPERS (512x512)** | **HOUSE (512x512)** |

Fig 4.9: Example of Inpainting with Colored Images.

| **Images** | **MSE** | | | **PSNR** |
|---|---|---|---|---|
| | **R** | **G** | **B** | |
| Lena | 12.11 | 10.78 | 12.07 | 37.46 |
| Peppers | 21.76 | 18.49 | 17.60 | 35.27 |
| House | 2.18 | 14.78 | 15.02 | 37.85 |

Table 4.9: Individual R,G,B values of MSE and PSNR of the Color Images

**4.3 Limitations of the algorithm**

Images having missing pixels or cracks (of white background) in the border areas of the image couldn't be worked upon with this algorithm, because of the absence of pixels in the surrounding areas near the border of the image. In such cases, the algorithm couldn't search for pixels which it can utilize to inpaint the given image.

|  |  |
|---|---|
| **ORIGINAL IMAGE** | **IMAGE CONVERTED TO BLACK AND WHITE (PRESENCE OF WHITE BACKGROUND IN THE BORDER AREA)** |

Fig 4.10: Example showing an image which won't work with the algorithm

Also an image after pixel restoration, looks perfect to the human eyes until it is zoomed. Upon zooming the restored pixels can be easily seen. This is another limitation of the algorithm. The pixel restoration is not 100% perfect. Example of such cases are shown below. The image will seem to be perfectly restored with the original pixels, in the areas having the white mask, but upon zooming, the pixels can be seen to be uneven.

| **Restored Images (Seems to be perfect)** | **Zoomed area shows uneven texture of the restored pixel** |
|---|---|
|  |  |
|  |  |

Fig 4.11: Example of Restored Images with their Zoomed View to show the restored pixels

# CHAPTER 5
# CONCLUSION

---

Conventional strategies are primarily pixel-based; their adequacy is profoundly related with the original information, and they face limitations in repairing images with sporadic compositions. In the frequency domain, our strategy can viably beat this disadvantage. This paper analyzes coding algorithm of JPEG image and proposes a K-Nearest Neighbor (KNN) approach to perform inpainting in the DCT Coefficients to get a more optimized compression ratio. Discarding DCT coefficients is generally done in image or video compression, in any case, the procedure is not followed; since in the event that we had, some high frequency information would be lost, and the picture quality would go down to some degree.

We utilize DCT rather than discrete fourier change (DFT) in this paper for four reasons:
1) DFT decomposes the picture into intensity and frequency components, but due to the lack of frequency components in the image position information, the spatial attributes of the image are not good;
2) DCT is is real-valued, while DFT produces complex values that include the magnitude and phase, which are more hard to actualize than DCT;
3) DCT has great directionality, this makes the textures and details of interest in different direction can be well extracted;
4) Both DFT and DCT have fast algorithms, taking DCT into consideration doesn’t lead to much time utilization contrasted with pixel-based techniques.

The basic idea is to detect the mask with absolute white background to recreate the image in the frequency domain. Contrasted with previous approaches, our method can be well inpaint images.

### 5.1 Future Scope

The method can be further studied and improved to work efficiently with images having missing pixels or cracks in the borders of the image. Also images having missing pixels from a large area of the image concentrated on the same location couldn't be used. The method failed to give any output when given an image of the above mentioned type. Secondly it also failed with images of certain size and shape. It had to be of perfect size preferably of 512x512 to get the best optimum output. Third, the compression algorithm used couldn't produce color images and always returned gray scale images in the end which shouldn't have happened in case of the baseline JPEG algorithm used. The code needed further modifications. While the inpainting algorithm was working fine both with color and gray scale images.

It should be noted that a high PSNR value is produced by the propinquity between the inpainted one and its original, but only high PSNR values doesn’t mean better image quality. As shown in Figs. 4.11,  made similar pixel values to the original image and reached a high PSNR. However, they weren’t able to inpaint the images because the structure wasn’t restored, and the destroyed part was still out of shape. So only this study method can achieve good results both in the objective PSNR and in the subjective, being visually satisfactory or acceptable.

# CHAPTER 6
# REFERENCES

[1] Li, Q., Han, Y., & Dang, J. (2014). Image decomposing for inpainting using compressed sensing in DCT domain. *Frontiers of Computer Science , Volume 8, Issue 6 , pp 905-915* , 905-915.

[2] Christine Guillemot, Mehmet Turkan, Olivier Le Meur, Mounira Ebdelli (2013).Image Inpainting Using LLE-LDNR And Linear Subspace Mappings. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference*,(pp- 1558 - 1562) Vancouver.IEEE

[3] Zhiqianga, L., Xiaoxin, S., Changbin, D., & Qun, D. (21-23 Sept. 2013 ). JPEG Algorithm Analysis and Application in Image Compression Encryption of Digital Chaos. *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2013 Third International Conference* (pp. 185 - 189 ). Shenyang: IEEE.

[4] Vahid Bastani, M. S. (2010). Image compression based on spatial redundancy removal and image inpainting. *Journal of Zhejiang University SCIENCE C, Volume 11, Issue 2 ,* , 92-100 .

[5] Yang, E.-h., & Wang, L. (2007). Joint Optimization of Run-Length Coding, Huffman Coding and Quantization Table with Complete Baseline JPEG Compatibility. *Image Processing, 2007. ICIP 2007. IEEE International Conference on (Volume:3 )* (pp. III - 181 - III - 184 ). San Antonio, TX : IEEE.

[6] Awate, S., & Whitaker, R. (2006). Unsupervised,Information-Theoretic,adaptive image filtering for image restoration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:28 , Issue: 3 )* , 364 - 376 .

[7] Shih, T., Chang, R.-C., Lu, L.-C., & Ko, W.-C. (2004). Adaptive Digital Image Inpainting. *Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on (Volume:1 )* (pp. 71 - 76). IEEE.

[8] Lakhani G. (2003). Modified JPEG Huffman Coding. *Image Processing, IEEE Transactions on (Volume:12 , Issue: 2 )* , 159 - 169 .

[9] Mitchell A. Golner, W. B. (2002). Modified JPEG Image Compression With Region-dependent Quantization. *Circuits, Systems and Signal Processing, Volume 21, Issue 2 ,* , 163-180.

[10] Huynoh-Thu Q, G. M. (2008). Scope of validity of psnr in image/video quality assessment. *Electronics Letters (Volume:44 , Issue: 13 ), IET* , 800 - 801 .

[11] Wang, Z., Bovik, A.C., (2009). "Mean squared error: love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag., 26(1):98-117*.

[12] M. Bertalmio, G. S. (2000, July). Image inpainting. *ACMSIGGRAPH* , pp. 417– 424.

Weblink: http://www.jpeg.org/

[13] G,G.(2006). Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising —part I: adaptive. *IEEE Transactions on Image Processing* , *Vol 15*

[14] G,G.(2006). Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising —part II: adaptive. *IEEE Transactions on Image Processing* , 555–571.

[15] A. Wong and J. Orchard (2006), "A nonlocal-means approach to exemplar-based inpainting," in *IEEE Int. Conf. Image Process. (ICIP)* , pp. 2600–2603.

[16] Telea, A. (2004). An image inpainting technique based on the fast marching method. *J. Graphics Tools, vol. 9, no. 1* , pp. 25–36.

[17] Tschumperle, D. (2006, March 01). Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *Springer* , pp. 65-82.

[18] N. Ahmed, T. N. (Jan 1974). Discrete Cosine Transform. *IEEE Trans. Commun. 23* , 90–93.

[19] Leung, A. E. (1999). Texture synthesis by non-parametric sampling. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on (Volume:2 )* (pp. 1033 - 1038 vol.2 ). Kerkyra: IEEE.

[20] A. Criminisi, P. P. (Sept 2004). Region filling and object removal by examplar-based inpainting. *IEEE Trans. on Image Processing, vol. 13, no. 9* , 1200 – 1212.

[21] C. Barnes, E. S. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graphics, vol. 28, no. 3* .

[22] Y. Wexler, E. S. (July 2004). Space-time video completion. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on (Volume:1 )* (pp. I-120 - I-127 Vol.1 ). IEEE.

[23] Sun, Z. X. (May 2010). Image inpainting by patch propagation using patch sparsity. *IEEE Trans. on Image Processing, vol. 19,no. 5* , 1153–1165.

[24] C. Studer, P. Kuppinger, G. Pope, and H. Bolckei (May 2012), "Recovery of sparsely corrupted signals," *IEEE trans. on Information Theory*, vol. 58, no. 5, pp. 3115 – 3130.

[25] Mitchell A. Golner, Wasfy B. Mikhael, and Venkatesh Krishnan (2002), Modified JPEG Image Compression with region-dependent quantization, *Springer*

[26] Gopal Lakhani, Modified JPEG Huffman Coding, *IEEE Transactions On Image Processing, Vol. 12*

[27] Timothy K. Shih., Rong-Chi Chang, Liang-Chen Lu, Wen-Chieh Ko and Chun-Chia Wang, Adaptive Digital Image Inpainting,Proceedings of the *18th International Conference on Advanced Information Networking and Application (IEEE)*

[28] Suyash P. Awate,Unsupervised Information-Theoretic Adaptive Image Filtering for Image Restoration,*IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 28, NO. 3*.

[29] En-hui Yang and Longji Wang, Joint Optimization Of Run-Length Coding, Huffman Coding And Quantization Table With Complete Baseline Jpeg Compatibility, *ICIP 2007 (IEEE)*

[30] Vahid Bastani, Mohammad Sadegh, Keyvan Kasiri, Image Compression Based On Spatial Redundancy Removal And Image Inpainting, *Springer,2010*

Weblinks:

[31] http://www2.irccyn.ec-nantes.fr/ivcdb/

[32] http://www.imageprocessingplace.com/root_files_V3/image_databases.htm

[33] http://decsai.ugr.es/cvg/CG/base.htm

[34] http://mict.eng.u-toyama.ac.jp/mictdb.html

# CHAPTER 7
# APPENDIX

---

## Abbreviations Used:

| | |
|---|---|
| DCT: | Discrete Cosine Transform |
| FDCT: | Forward DCT |
| IDCT: | Inverse DCT |
| KNN: | K-Nearest Neighbor |
| KLHT: | Karhunen–Loeve–Hotelling Transform |
| JPEG: | Joint Photographic Experts Group |
| PDE: | Partial Differential Equation |
| NLM: | Non-Local Means |
| IBL: | Instance Based Learning |
| RLE: | Run-Length Encoding |
| CS: | Compressed Sensing |
| LLE-LDNR: | Locally Linear Embedding with Low-Dimensional Neighborhood Representation |
| NLM: | Non-Linear Matrix |
| UINTA: | Unsupervised Information Theoretic Adaptive Filter |
| PSNR: | Peak Signal-To-Noise Ratio |
| SSIM: | Structural Similarity Index Measurement |
| MSE: | Mean Square Error |

# CHAPTER 8
# PUBLICATIONS

---

**Published Paper:**

[1] Biprajit Bhattacharjee, Sanyam Anand, Usha Mittal (2015). A novel KNN approach to implement inpainting and compression/decompression technique and get an optimized output with the help of DCT Coefficients, *International Journal of Computer Science and Technology, vol. 6,issue1 ,Spcl 1, pp-57-59*

**Sent for Review:**

[2] Biprajit Bhattacharjee, Sanyam Anand, Usha Mittal (2015). Image Inpainting using KNN in DCT Domain and Compression to get an optimized output, IJAER