SECURE IMAGE ENCRYPTION USING CHAOTIC MAPS AND STEGANOGRAPHY

Thesis submitted to the SASTRA Deemed to be University in partial fulfilment of the requirements for the award of the degree of

B. TECH ECE(CPS)

ECE 400: MAJOR PROJECT

Submitted by

CHAKKA SAI VENKAT

(Reg.No:124160012)

SHASTRULA SAAKETH SHARMA

(Reg.No:124160081)

April 2024

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR, TAMILNADU, INDIA-613 401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR, TAMILNADU, INDIA-613 401

Bonafide Certificate

This is to certify that the thesis titled "**Secure Image Encryption using Chaotic maps and Steganography"** submitted as a requirement for the course. **ECE400: MAJOR PROJECT** for B. Tech ELECTRONICS & COMMUNICATION ENGINEERING program, is a bonafide record of the work done by **Mr. Shastrula Saaketh Sharma (Reg.No:124160081) & Mr. Chakka Sai Venkat (Reg.No:124160012)** during the academic year 2023-2024, in the school of ELECTRICAL & ELECTRONICS ENGINEERING, under my supervision.

Signature of the Project Supervisor: 24/4/24

Name with Affiliation: Dr. Lakshmi .C**,** AP-III, SEEE

Date:

Project viva-voce held on: _____________

Examiner 1 **Examiner 2**

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613401

Declaration

We declare that the thesis titled "**Secure Image Encryption using Chaotic maps and Steganography"** submitted by us is an original work done by us under the guidance of **Dr. LAKSHMI.C, Asst.Professor-III, School of Electrical and Electronics Engineering, SASTRA Deemed to be University** during the sixth semester of the academic year 2023-2024, in the school of Electrical and Electronics Engineering. The work is original and wherever we have used materials from other sources, we have given due credit and cited them in the text of the thesis. This thesis has not formed the basis for the award of any degree, diploma, associate-ship, fellowship, or other similar title to any candidate University.

Signature of the Candidates: Ch. Sai Venkat

Saaketh sharma

Name of the candidates: CHAKKA SAI VENKAT

SHASTRULA SAAKETH SHARMA

Date:

ACKNOLEDGEMENTS

First of all, we express our gratitude to **Dr. S Vaidhya Subramaniam,** Vice Chancellor, SASTRA Deemed to be University, who provided all facilities and constant encouragement during my study. We extend our sincere thanks to **Dr. R Chandramouli,** Registrar, SASTRA Deemed to be University for providing the opportunity to pursue this project.

We dedicate our whole hearted thanks to **Dr. K. Thenmozhi**, Dean (SEEE), SASTRA Deemed to be University for the moral support. We also thank **Dr. K. Sridhar**, Associate Dean (ECE), for his encouragement.

We would like to express our deepest gratitude to our guide **Dr. Lakshmi .C,** AP-III, SEEE for her valuable suggestions and guidance throughout the project. a special thanks to our panel members **Dr. Amritharajan .R**, Professor, SEEE and **Dr. Nithya . C**, Asst.Professor-III, SEEE for rectifying our mistakes and correcting them and mentoring our project to complete in a better way.

We also take this opportunity to thank our parents for their moral support during this project.

ABSTRACT

Technology is undergoing a rapid development. Advancements in various applications such as defence, healthcare service, online services etc... lead to a huge data storage. So, ensuring data privacy is very important. This highlights the importance of steganography and image encryption. One of the primary difficulties caused with the image encryption is the reverse process, which is decryption part. The quality of the decrypted image may decrease due to the various techniques used for encryption part.

This project clearly discusses about how the color image is being encrypted using the 1D chaotic map and how the Stego-Encrypted image is generated from the input data taken from the user. Initially a input color image of 256*256 size is taken. This color image is undergone through the encryption using the simple logistic map. Then the confusion and diffusion processes are done for increasing security. So as the encrypted image is of the type RGB form, it is converted to YCbCr form. Y component is extracted from it and it is divided into 8*8 non-overlapping blocks. Each block is undergone through the Discrete Cosine Transform (DCT) process and Quantization is applied to each block. On the other side, data is taken as input from the user. Run Length encoding technique is used to encode the input text. The stream of binary data generated is utilized for embedding into quantized image. As the pixel is of 8-bit binary data, the last 3 bits of each pixel are replaced with the first 3 bits of run length encoded bits and so on. So, finally the stego-encrypted image is generated. The result will show the stego-encrypted image which has a embedding capacity of 1,96,608 bits.

After the generation of stego-encrypted image the reverse process is carried out as IDCT is performed on the stego-encrypted image. Now the decoding of text message is done using run length decoding technique and the text extraction is carried out. Then Y channel is combined with CbCr color space then this YCbCr image is converted to RGB color space, and the original image is recovered.

This project will result in the stego-encrypted image and the recovery of original image from the stego-encrypted image.

Key Words**:** *Logistic Map, DCT, Quantization, Run-length Encoding, Stego-Encrypted Image.*

Specific Contribution: 1D chaotic encryption and decryption are done. DCT and Quantization processes are done to increase the security and run length encoding process is done for the text data and embedded into quantized image.

Specific Learning: Acquired good knowledge on how DCT and Quantization processes work and how the encryption is done and learnt about the run-length encoding and how to embed data into image.

24/4/24

Signature of Guide: Reg No : 124160012 , 124160081

Name: Dr. Lakshmi .C**,** AP-III, SEEE Name: CHAKKA SAI VENKAT

SHASTRULA SAAKETH SHARMA

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVATIONS

RGB- Red, Green, Blue

YCbCr- Y-Luminance, Cb-Chrominance Of Blue, Cr-Chorminance Of Red

DCT- Discrete Cosine Transform

IDCT- Inverse Discrete Cosine Transform

ASCII- American Standard Code for Information Exchange

CHAPTER-1

INTRODUCTION

In today's world we use images a lot for our daily life. Images play a crucial role in our daily life communication. Images play a crucial role in various sectors such as defense, healthcare, and online services etc.... With this broad usage of images, we require a safe transmission of information. Securing the Privacy and Security of image data is a major concern. So, we use Encryption and decryption techniques to protect sensitive image information.

Due to this large usage of technology for transfer of data we need a medium for secure transfer of data. So, only the sender and receiver can know the information that is being transferred. Here, comes the encryption which makes that possible by converting the data into a secret code and only the person who knows that secret code can decrypt that and can obtain the data.

Chaotic maps are mathematical functions with unpredictable behavior. Chaotic maps are used in encryption algorithms to enhance the safety of images and image information. Chaotic maps are used for key generation by which unpredictable keys are generated for encryption and decryption. Chaotic maps are also used for Scrambling, Confusion, and diffusion processes. It can also be used in pixel transformation by which the pixel values are transformed so that it makes it difficult for attackers to retrieve the original image.

1D chaotic map is a mathematical function which exhibits its behavior in one dimension. The 1D logistic map is used as a chaotic map which generates random numbers which can be further used as encryption and decryption keys.

The encryption process involves confusion and diffusion techniques. In confusion we will change the letters in the text by following a certain mathematical function and a key, we cannot decode it without knowing the proper key. Diffusion will change the order of the text the one change in one letter makes the changes in all letters making it difficult to decode.

The Y component extracted from the YCbCr image is divided into 8*8 non overlapping blocks the image is converted into 8*8 non overlapping blocks because the division of the image into small blocks makes it simpler to analyze the image each 8*8 block focusses on the small details in the picture by removing the less important information from the image when DCT and quantization is performed to compress the image which saves the space for storage and which also reduces the computational complexity.

The Discrete Cosine Transform (DCT) is a mathematical function which is used to convert the data from the image to frequency domain. It represents the data as high frequency and low frequency components. DCT takes an 8*8 block image as input and converts it into different frequency coefficients for analysis.

The DCT process helps in making quantization. Quantization is used to compress the image and audio data. It is used in image compression, in image encryption the DCT is performed before the quantization which helps in reducing the number of bits to represent the data. When we perform DCT the components with high frequency or high DCT coefficients are removed mostly while performing quantization because they contain less important information.

CHAPTER 2

LITERATURE SURVEY AND OBJECTIVE

Iram sabha and Shabir A. parah proposed the scheme of encryption scheme based on convolutional encoder and logistic map in which the input image is given to encoder convolutional network then the image is processed for encryption process and then the image is recovered by decryption and decoder convolutional network and then the image is recovered as the original image. This joint compression technique reduces the bandwidth and storage requirement for the images. The higher dimensional image is taken as input and the chaos-based encryption is done and the lower dimensional image is obtained as output. The quality of the image is decreased by using joint compression technique.[1]

Khalid and Sara proposed a methodology in which image encryption is done based on scrambling and diffusion process. First the image is divided inti three channels as R, G, and B. and these three channels are divided into sub images and blocks. The scrambling and diffusion process make changes in that sub divided images and uses logistic map to obtain the encrypted image.[2]

Mostafa and Khalid have applied multiple methods to encrypt the image and to secure the data text embedded in the image. They mentioned the hyperchaotic process by which the image is scrambled, and confusion and diffusion process takes place. The image is taken as an RGB color model and then it is converted to YCbCr color space. After converting it to YCbCr model the Y component is extracted and the 256*256 image is divided into 8*8 non overlapping blocks then each block is performed with DCT and quantization process by which the image compression takes place then the pixel value of each block is taken and LSB is erased and the secret message is performed with Huffman coding then this secret message is embedded in that LSB and the encrypted message is generated.[3]

Ahmed, Hussein and Hamed gave us a technique in which we use three maps for encryption. First the image is divided into three color channels as R, G and B. Three distinct chaotic maps-Baker , Arnold and Henon maps are used for three different R-G-B channels for encryption, usage of three maps increases the computation complexity as we need to generate 3 keys and it also increase the security as we require 3 keys for decryption. As we use 3 distinct maps vulnerabilities also increases as they are different for different maps.[4]

Benxuan Wang and Kwok-Tung Lo proposed a model in which they combined encryption and compression concepts using deep learning concepts. It enhances high security by swapping logistic map with a key. It achieves high security by having good compression efficiency and it avoids separate key transmission. But it is sensitive for changes in the image and it increases the computational load by causing additional permutations in the image.[5]

Xiuli Chai, Xianglong Fu, and Zhihua Gan proposed a method on efficient chaos-based image compression and encryption technique using block compressive sensing. In this they used discrete wavelet transform(DWT) to split the images into blocks instead of DCT. The technique uses block compressive sensing compresses the blocks with varying ratios based on importance using chaotic map. It uses SHA-256 hash map algorithm for image encryption. It helps in efficient encryption and compression and by using permutations and chaos it have enhanced security. But by using these many techniques it has loss of image quality and for the same it has high computational complexity.[6]

Rupali Bhardwaj and Divya Khanna proposed a method in which it enhances the safety of image steganography by means of combining cryptography and steganography. Rather than at once hiding message bits in a cowl image, the technique first scrambles the message using a 2D Arnold Cat Map after which encrypts the message earlier than embedding it into the cover photo using basic LSB approach. This technique provides an extra layer of protection and complexity, making it harder for unauthorized customers to hit upon the hidden message. The technique is proven to have better overall performance than simple LSB, with higher PSNR and lower MSE, making sure the best of the stego-photograph remains high. However, the approach may be computationally in depth because of using image scrambling techniques.[7]

In their paper, Dr. R. Sridevi, Vijaya Lakshmi Paruchuri, and K.S. SadaSiva Rao suggest a way that combines steganography and cryptography to hide secret statistics in pictures. The technique involves embedding information the use of the Least Significant Bit (LSB) method and encrypting the ensuing photo with the Advanced Encryption Standard (AES). The encrypted image can then be decrypted by means of the receiver to get better the original image and extract the hidden information. The authors display the effectiveness in their method through experimental consequences, highlighting its capability for steady communique.[8]

The proposed picture encryption and steganography scheme by means of Hongmei Tang, Gaochan Jin, Cuixia Wu, and Peijiao Song gives numerous advantages. Firstly, it combines grey price substitution and role permutation, leveraging the chaotic logistic map, to acquire a excessive degree of safety. The use of chaos principle and cryptography enhances the encryption process, making it touchy to preliminary situations and pseudo-random, as a result resisting attacks based totally on statistical evaluation. Additionally, the scheme employs matrix coding in steganography, enhancing the embedding efficiency appreciably. However, there are some limitations to recollect. The scheme's reliance on chaotic sequences makes it computationally intensive, probably impacting performance in actual-time programs. Furthermore, the scheme's security depends closely on the secrecy of the preliminary values of the chaotic logistic maps, which might be difficult to manipulate in exercise.[9]

The proposed methodology for photo steganography using the LSB algorithm by using S. Sravani and R. Ranjith entails hiding a secret image inside a cover photograph and shifting it over the net. The system includes converting RGB pix into grayscale, extracting pixel values, encrypting using LSB wavelet set of rules, and reading overall performance thru MSE and PSNR values. The advantages of this approach lie in its potential to hide facts effectively, mainly for security-based totally packages. However, it may be computationally extensive and liable to records loss because of outside noise. Sravani and Ranjith's technique gives a sensible solution for secure facts communique, specially in protection areas, but similarly enhancements could enhance its performance and robustness.[10]

The proposed method, CIEST, combines cryptography and steganography to decorate the security of virtual snap shots. It encrypts the name of the game image the usage of a two-degree method, inclusive of DNA encoding, after which hides the encrypted image within the cowl photo. Multiple mixed chaotic maps are utilized to enhance the randomness and robustness of the set of rules. The advantages of CIEST include high safety, high embedding capability, and resistance to various attacks. However, it is able to have some negative aspects which include improved computational complexity due to the use of a couple of chaotic maps. Overall, CIEST offers a unique approach to photograph safety, as proposed by H. Aparna and J. Madhumitha from the Department of ECE, College of Engineering, Guindy, Anna University, Chennai, India.[11]

OBJECTIVE

Objective: Develop a system for encrypting color pics the usage of a 1D chaotic map, applying confusion and diffusion tactics for security, changing the picture to YCbCr format, dividing it into 8x8 blocks, making use of DCT and quantization, and embedding text the usage of run-length encoding. The system should also encompass a decryption technique to get better the unique image from the stego-encrypted image.

CHAPTER 3

METHODOLOGY

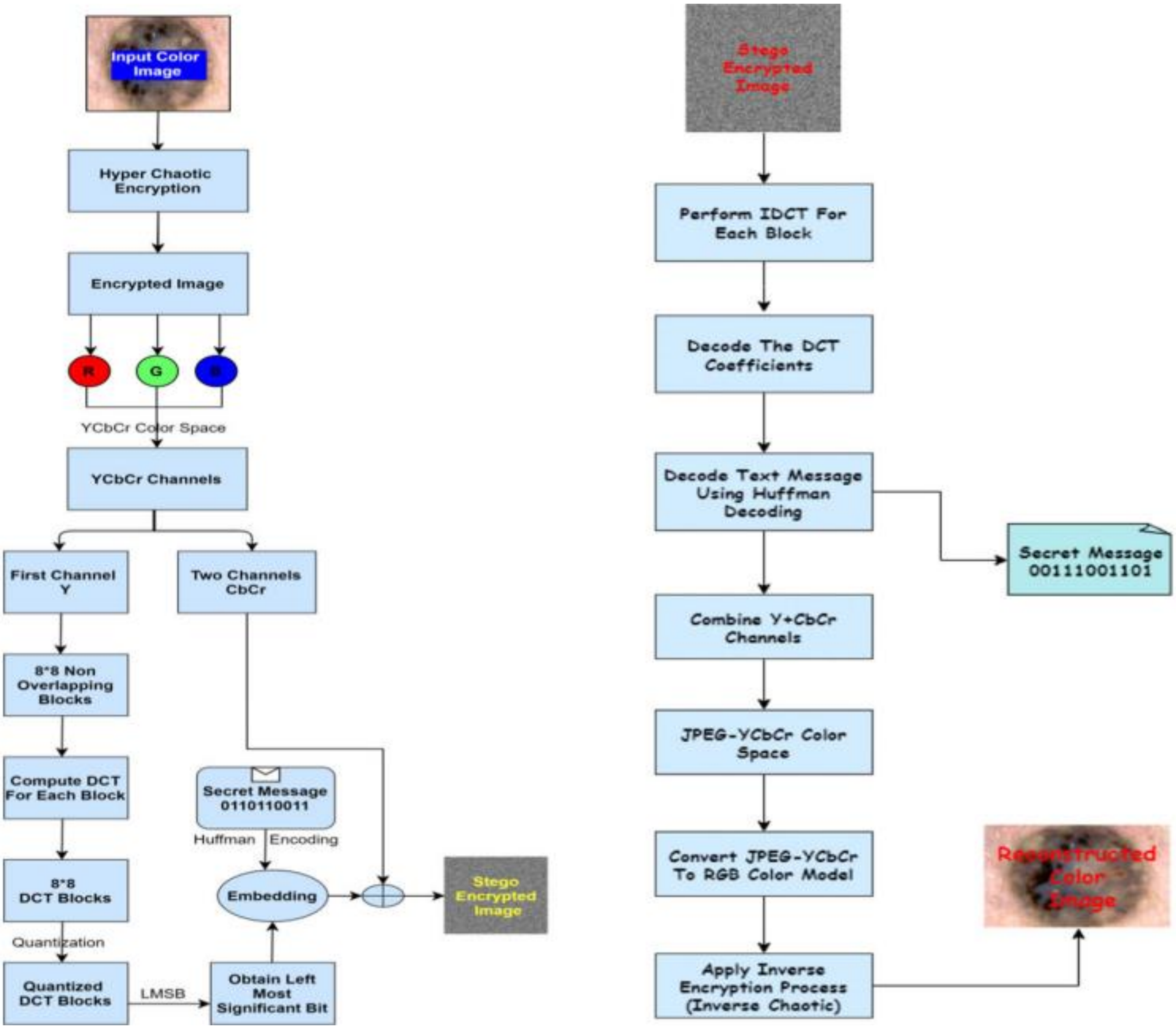

Input Color Image
Hyper Chaotic Encryption
Encrypted Image
R
G
B
YCbCr Color Space
YCbCr Channels
First Channel Y
Two Channels CbCr
8*8 Non Overlapping Blocks
Compute DCT For Each Block
8*8 DCT Blocks
Quantization
Quantized DCT Blocks
LMSB
Obtain Left Most Significant Bit
Secret Message 0110110011
Huffman Encoding
Embedding
Stego Encrypted Image
Stego Encrypted Image
Perform IDCT For Each Block
Decode The DCT Coefficients
Decode Text Message Using Huffman Decoding
Secret Message 00111001101
Combine Y+CbCr Channels
JPEG-YCbCr Color Space
Convert JPEG-YCbCr To RGB Color Model
Apply Inverse Encryption Process (Inverse Chaotic)
Reconstructed Color Image

Fig.3.1 Methodology

This Block Diagram demonstrates the methodology.

- The color image is taken as the input.
- 1D chaotic encryption is performed on the image which is taken as the input.
- After encryption the image which is in RGB color space is converted to YCbCr color space.
- After converting it to YCbCr color space the image is classified as Y channel and CbCr channel.
- The Y component is extracted, and further operations are performed on this Y component image.
- The Y component image extracted from YCbCr image is divided into 8*8 non overlapping blocks.
- After converting the image into 8*8 non overlapping blocks DCT is performed for each block
  DCT- Discrete Cosine Transform is a technique in which the image is converted into frequency domain. The information is stored as the high frequency components whereas low frequency is slightly unnoticed. This DCT is commonly used for image compression technique in which low frequency components are filtered as they are used to store less important information.
- After performing DCT quantization is applied to the image.
  Quantization- The Quantization matrix is used to round off the DCT coefficients to nearest integer values which reduces precision of the image, this process helps in image compression. After performin g quantization encoding techniques is used to encode the data into the image.
- Next a text message is taken as input.
- This text message is converted to ASCII values then this ASCII value is converted to binary values (0's & 1's).
- A run length encoding technique is used to encode this binary data.
- After performing this run length encoding the obtained encoded data is divided into a set of 3 bits so that this data can be embedded into the pixel values of the DCT and Quantized image.
- The stego encrypted image is generated.
- After obtaining stego encrypted image the reverse process is carried out.
- The reverse process starts with performing IDCT for the stego-encrypted image.
- Then the text is decoded using decoding technique.
- After obtaining the text the YCbCr image is obtained.
- Then the YCbCr image is converted into RGB color space.
- Now the reconstructed (Original) image is obtained.

CHAPTER 4

WORK CARRIED OUT

4.1 1-D Chaotic Encryption and Converting RGB image to YCbCr Image

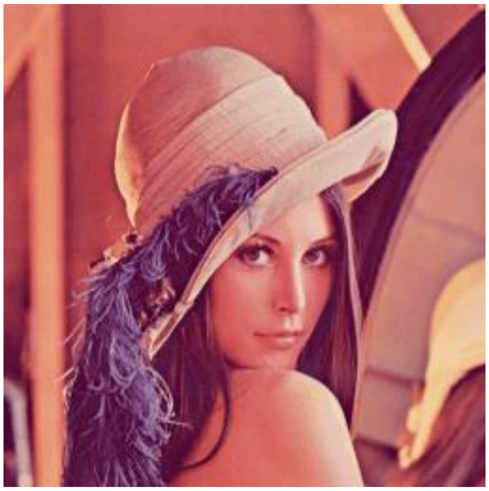


Fig 4.1.1 – Input Image

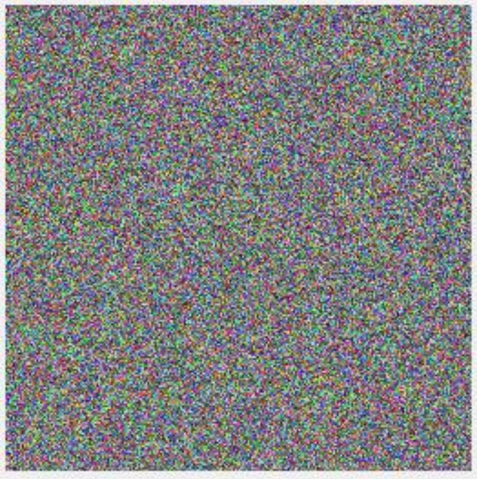


Fig 4.1.2 – Encrypted Image

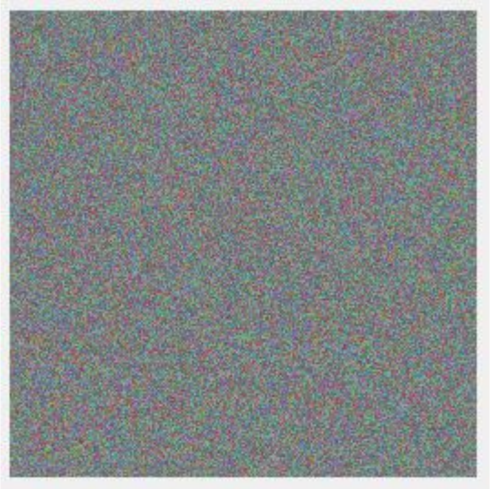


Fig 4.1.3 – YCbCr Image

- Initially, the formula of logistic map is considered, that is $x_{n+1} = LM(m, x_n) = m * x_n * (1 - x_n)$
- The pixel values of the input image are stored in an array. Based on the logistic map formula, chaotic sequence is generated. This process is called Confusion.
- For the diffusion process, sorting of pixel values are done with storing the sorting_Indexes, which can be helpful for decryption process retrieval.
- The array is again reshaped in the form of size of input image. Hence the encrypted image is generated.
- As we know that the encrypted image is in the form of RGB form, to convert into YCbCr form, we can use the formula ycbcr_image=rgb2ycbcr(encryptedImage)
- Now from the YCbCr image, Y- Component image and CbCr image as extracted separately using the formula, Y_comp=ycbcr_image(: , : , 1)
- Now, from the YCbCr image, Y component alone is extracted using the command

Y_comp = ycbcrImage(: , : ,1);

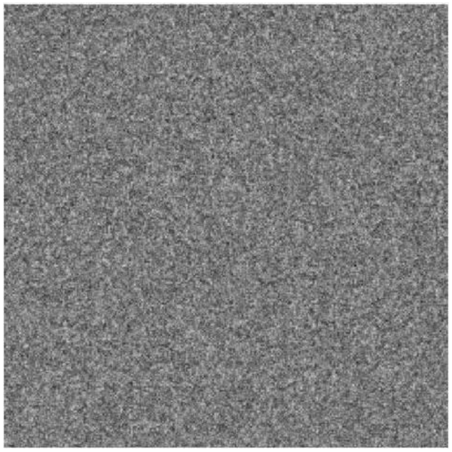


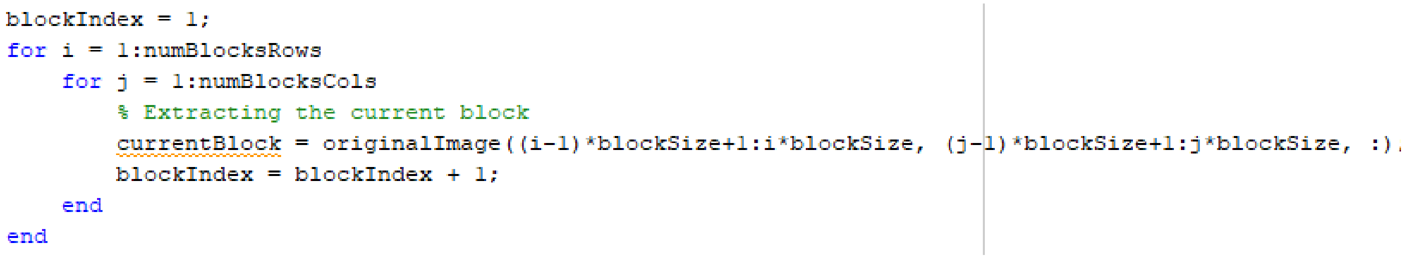
Fig 4.1.4 – Y_component Image

4.2 Dividing Image into 8 x 8 Non-overlapping Blocks

- So, this Y-component image which is 256*256 pixels image is being divided into 8*8 blocks. That means 32 blocks on each row and 32 blocks on each column.
- Initially we will create an array called blocks for storing all the 8*8 blocks and using subplot command we will showcase as output.

  Full_block = zeros (block_Size, block_Size, 3, num_Rows * num_Cols, 'uint8');

- The logic behind the dividing is that,



```
blockIndex = 1;
for i = 1:numBlocksRows
    for j = 1:numBlocksCols
        % Extracting the current block
        currentBlock = originalImage((i-1)*blockSize+1:i*blockSize, (j-1)*blockSize+1:j*blockSize, :),
        blockIndex = blockIndex + 1;
    end
end
```

Fig 4.2.1 – Logic behind division of Y-component image to 8*8 blocks

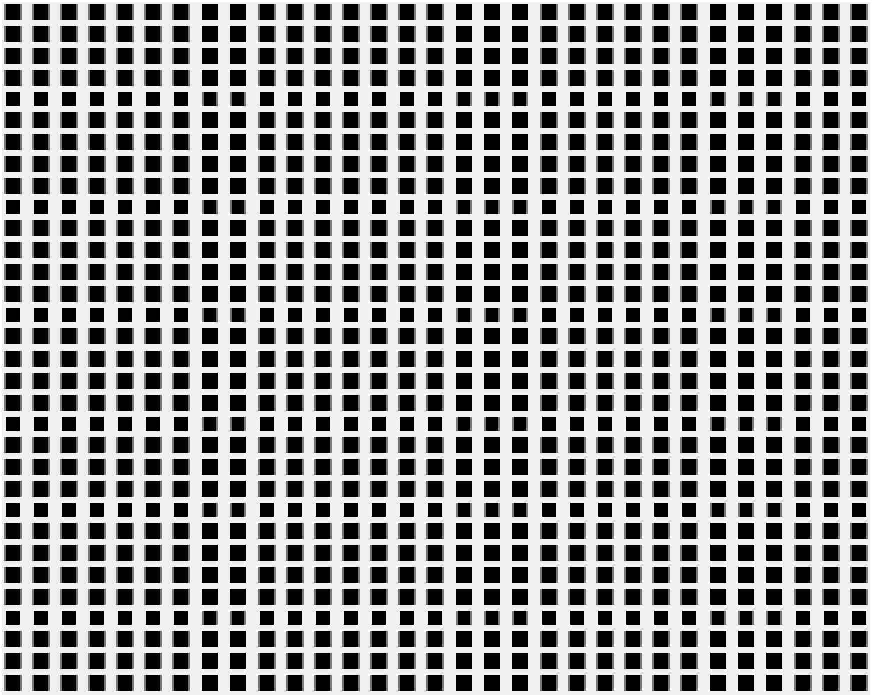


Fig 4.2.2 – Y-component image divided into 8*8 blocks

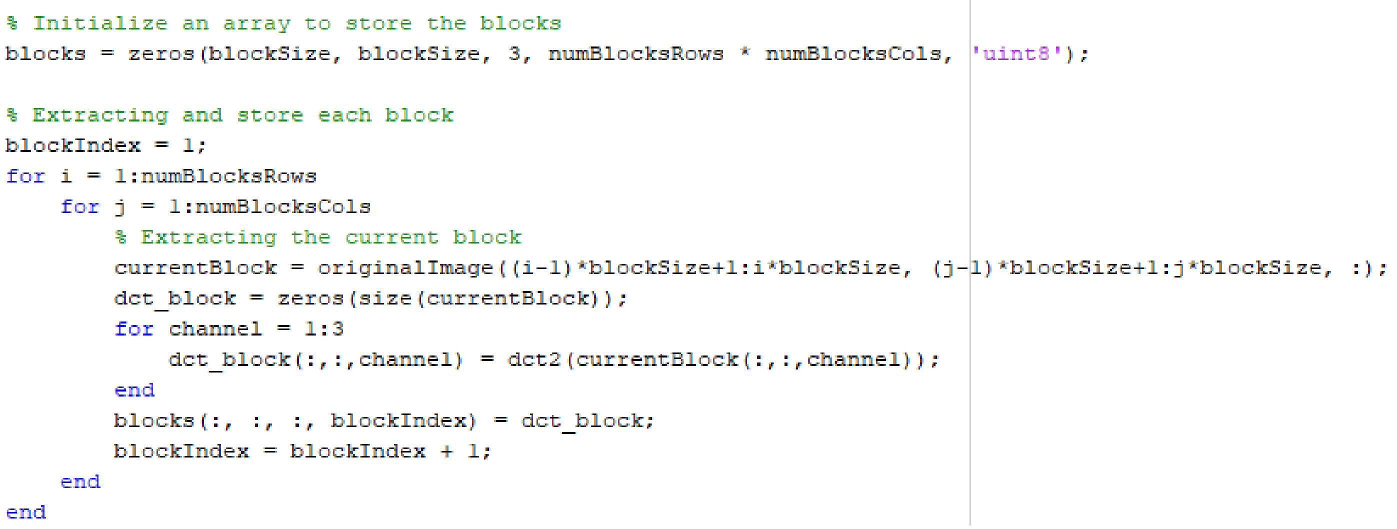
4.3 Applying Discrete Cosine Transform (DCT) for each non-overlapping blocks

- DCT which is Discrete Cosine Transform is an important process in the image encryption. The reason behind this is the Security and Robustness of the image. It is difficult for the attacker to decrypt the image as the transformed coefficients after applying DCT have less correlation value.
- As the randomness in the output image is high, so it can withstand with some of the popular attacks like brute force etc...
- The formula of DCT is as follows,

$$F(u,v) = \left(\frac{1}{4}\right) * c(u) * c(v) + \sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)^{*}\, cos[{}^{\pi}/_{8}\left(x+\frac{1}{2}\right)u]\ cos[{}^{\pi}/_{8}\left(y+\frac{1}{2}\right)v]$$

- As the pixel values of each block will be of the range 0 to 255. But to apply DCT , we it should be in the range of -128 to 127. So each pixel value is subtracted with -128. Then the DCT formula is applied to each block.
- DCT formula calculates set of 8*8 that means 64 coefficients in frequency domain. In Matlab we have a command called

DCT_block = dct2 (current_block)



```
% Initialize an array to store the blocks
blocks = zeros(blockSize, blockSize, 3, numBlocksRows * numBlocksCols, 'uint8');

% Extracting and store each block
blockIndex = 1;
for i = 1:numBlocksRows
    for j = 1:numBlocksCols
        % Extracting the current block
        currentBlock = originalImage((i-1)*blockSize+1:i*blockSize, (j-1)*blockSize+1:j*blockSize, :);
        dct_block = zeros(size(currentBlock));
        for channel = 1:3
            dct_block(:,:,channel) = dct2(currentBlock(:,:,channel));
        end
        blocks(:, :, :, blockIndex) = dct_block;
        blockIndex = blockIndex + 1;
    end
end
```

Fig 4.3.1 – Logic behind the DCT computation for each block

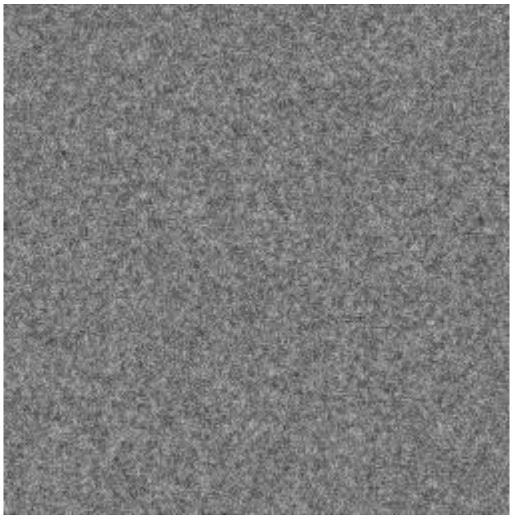


Fig 4.3.2 – Y-component Image

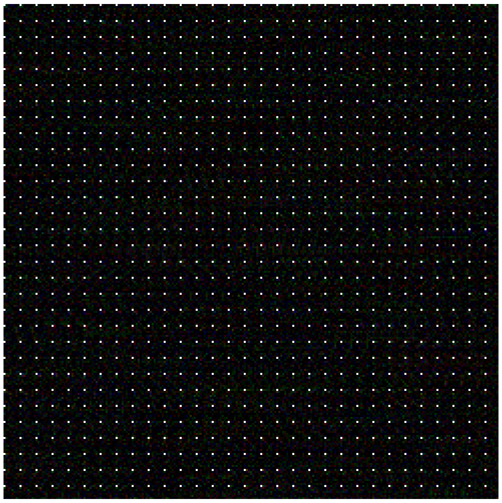
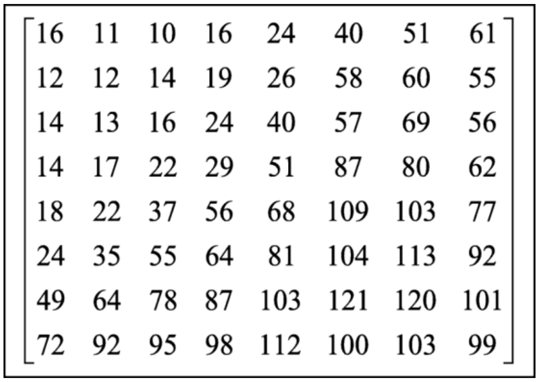


Fig 4.3.3 – DCT Image

4.4 Applying Quantization for each Block

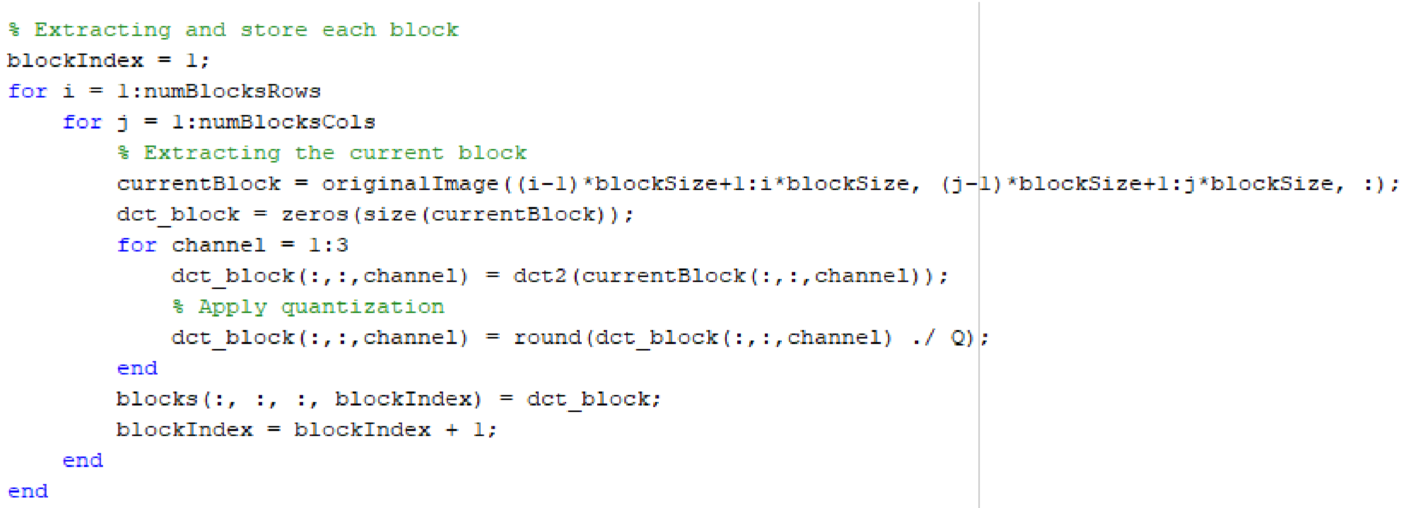
- Well Quantization is the method for image compression or image encryption used for reducing the precision of pixel values and mapping it to the smaller values and rounding off the coefficients using the standard quantization table.
- We use Quantization for compression and Encryption process, as this process introduces a noise in the transformed coefficients of the DCT image which makes the attacker harder to decode the image.
- Using Quantization, the DCT coefficients are scrambled and distorted and by using the Quantization matrix table, the coefficients are divided with matrix values and rounded off.
- The standard Quantization table,



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

Fig 4.4.1 – Quantization Table for JPEG Compression and Encryption

- Logic Behind the Quantization is that,



```
% Extracting and store each block
blockIndex = 1;
for i = 1:numBlocksRows
    for j = 1:numBlocksCols
        % Extracting the current block
        currentBlock = originalImage((i-1)*blockSize+1:i*blockSize, (j-1)*blockSize+1:j*blockSize, :);
        dct_block = zeros(size(currentBlock));
        for channel = 1:3
            dct_block(:,:,channel) = dct2(currentBlock(:,:,channel));
            % Apply quantization
            dct_block(:,:,channel) = round(dct_block(:,:,channel) ./ Q);
        end
        blocks(:, :, :, blockIndex) = dct_block;
        blockIndex = blockIndex + 1;
    end
end
```

Fig 4.4.2 – Logic for applying Quantization

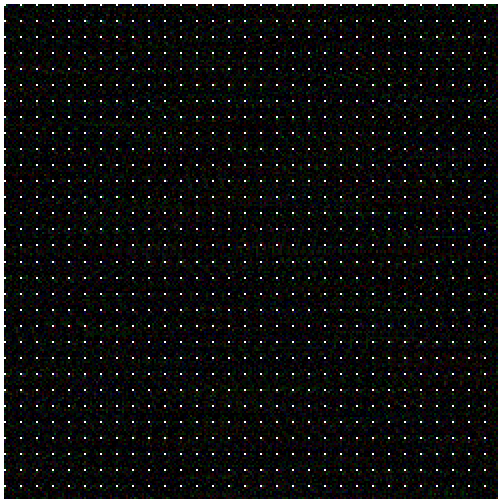


Fig 4.4.3 – DCT Image

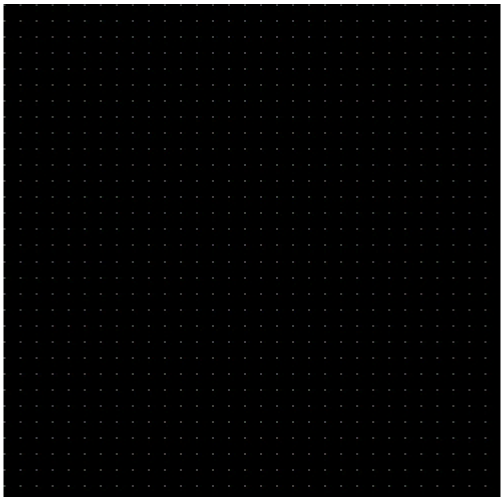
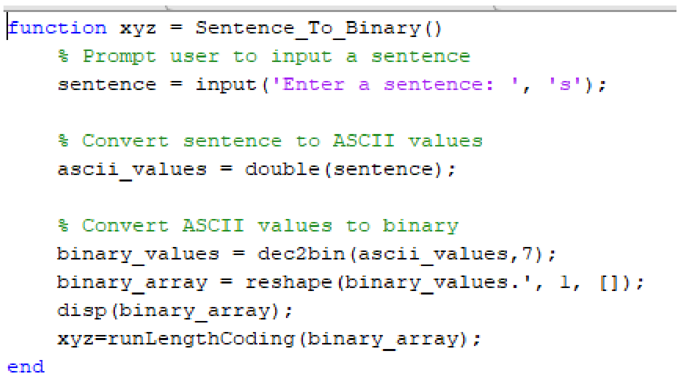


Fig 4.4.4 – Quantized Image

4.5 Converting Input Data to ASCII values and converting to Binary Stream of data

- Initially, input data is taken from the user. This data will be of string or array of characters.
- Each character has respective ASCII values. So based on character, ASCII values are written. This ASCII value is converted to 7-bit binary data.
- Finally the stream of binary data is formed and this data is stored in a binary_array for further steps.



```
function xyz = Sentence_To_Binary()
    % Prompt user to input a sentence
    sentence = input('Enter a sentence: ', 's');

    % Convert sentence to ASCII values
    ascii_values = double(sentence);

    % Convert ASCII values to binary
    binary_values = dec2bin(ascii_values,7);
    binary_array = reshape(binary_values.', 1, []);
    disp(binary_array);
    xyz=runLengthCoding(binary_array);
end
```

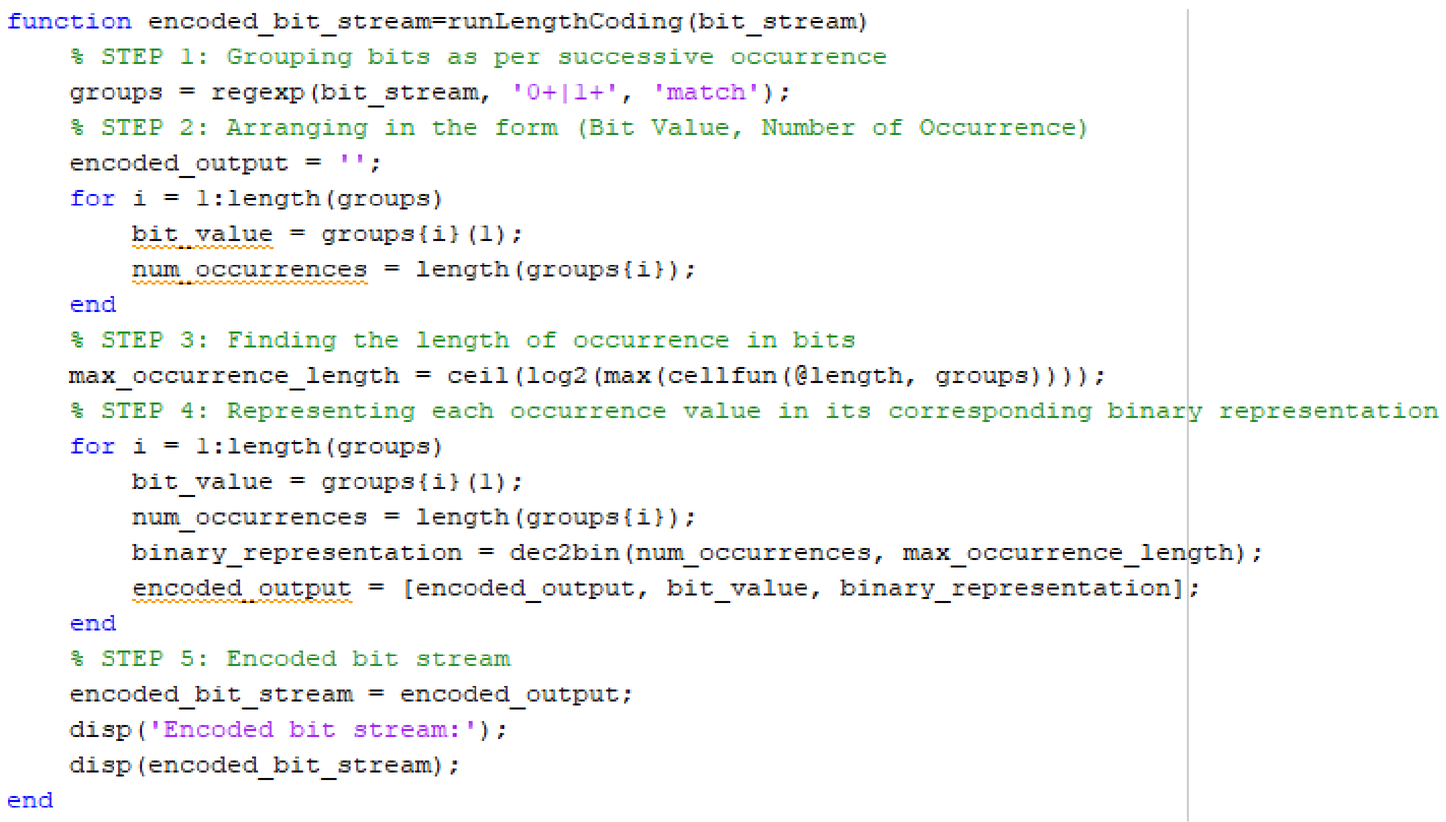
Fig 4.5.1 – Logic Behind Sentence to Binary

LETTER	ASCII VALUE	BINARY VALUE FROM ASCII VALUE
S	83	1010011
e	101	1100101
c	99	1100011
r	114	1110010
e	101	1100101
t	116	1110100
SPACE	32	0100000
M	77	1001101
e	101	1100101
s	115	1110011
s	115	1110011
a	97	1100001
g	103	1100111
e	101	1100101

Table 4.5.1 – How alphabetical letter is converted to Binary

4.6 Run Length Encoding

- The algorithm states that, splitting the input binary string based on the successive 0s and 1s.
- Then we have to count number of occurences of each splitted part along with the value 0 or 1. i.e., (BitValue, Number_of_Occurence)
- Take the maximum of all number_of_occurence values and store it in a variable called x.
- Find the value of p=$log_2(x)$. Now find ceil(p) value. Store it in a value, z=ceil(p).
- Now we have to convert each Number_of_occurence value to its binary value of length z.
- Represent the ordered pair as (BitValue, BinaryValue of Number_of_occurence).
- Combine all ordered pairs by removing brackets. This will be the algorithm for run-length encoding process.



```
function encoded_bit_stream=runLengthCoding(bit_stream)
    % STEP 1: Grouping bits as per successive occurrence
    groups = regexp(bit_stream, '0+|1+', 'match');
    % STEP 2: Arranging in the form (Bit Value, Number of Occurrence)
    encoded_output = '';
    for i = 1:length(groups)
        bit_value = groups{i}(1);
        num_occurrences = length(groups{i});
    end
    % STEP 3: Finding the length of occurrence in bits
    max_occurrence_length = ceil(log2(max(cellfun(@length, groups))));
    % STEP 4: Representing each occurrence value in its corresponding binary representation
    for i = 1:length(groups)
        bit_value = groups{i}(1);
        num_occurrences = length(groups{i});
        binary_representation = dec2bin(num_occurrences, max_occurrence_length);
        encoded_output = [encoded_output, bit_value, binary_representation];
    end
    % STEP 5: Encoded bit stream
    encoded_bit_stream = encoded_output;
    disp('Encoded bit stream:');
    disp(encoded_bit_stream);
end
```

Fig 4.6.1 – Run-Length Coding Logic

- >> run_length_main

  Enter the sequence to be encoded: 00000111110010000101

  Encoded bit stream:

  01011101001010010100100100011001

- Now, if we combine both the sentence to binary and run length encoding. The output will look like this :

  >> Sentence_To_Binary
  Enter a sentence: Chakka_and_Saaketh
  10000111101000110000111010111101011110000110111111100001110111011001001011111101001111
  00001110000111010111100101111010011010000
  Encoded bit stream:
  1001010011000001100100111010010010110001100100011100000110010001110001001010000111110
  10010110001101100011010001010010010100100011110000110010010110001001011010010110001100
  010001110000101001000111000000110010010101000011001001 1

4.7 Embedding the Binary Data into the Quantized Image to get Stego-Encrypted Image

- This process is divided into 2 main steps. First step is clearing the 3 LSB bits from each pixel value and Second step is dividing the stream of binary data released on the run length encoding part is divided into 3 bits. These 3 bits are to be replaced with the 3 LSB bits of each pixel to finally form the stego-encrypted image.
- **First Step : Clearing 3 LSB bits from each pixel**

  This can be achieved by doing AND gate operation between the pixel value and the complement of 7.

  Pixel & (~7)

  For example, let the pixel value be : 10110110

  Now, (10110110) & ~(00000111) = (10110110) & (11111000) = 10110000

- **Second Step : Dividing the stream of binary data into 3 bits and replacing with 3 LSB bits of pixel**

  This can be achieved using OR operation of pixel value and the 3 bits of run-length encoded data.

  Pixel | (x y z)

  For example, let the stream of data be : 100101001100000110010011101001001
  Let the pixel data be : 10111000

  Divide into 3: 100 101 001 100 000 110 010 011 101 001 001

  Now (10111000) | (100) = (10111000) | (00000100) = 10111100

CHAPTER 5

RESULTS AND DISCUSSION

This is the Sample Original Image which is taken as input. Encryption, Decryption, color space conversion, DCT, Quantization and embedded the text into this image is done in further processes.

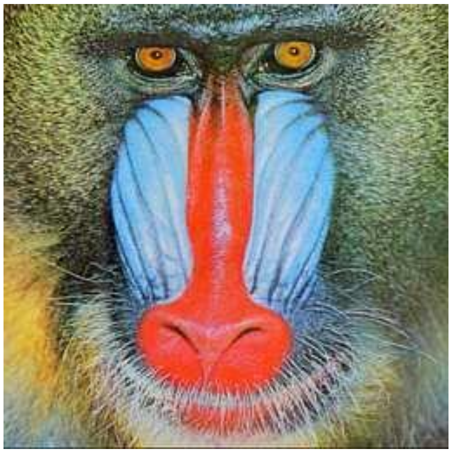
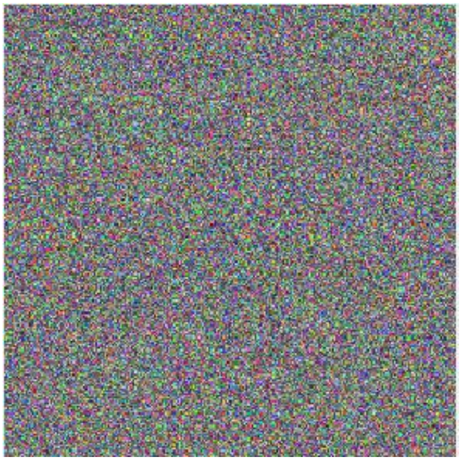


Fig.5.1.1 Original Image



This is the encrypted image generated after performing 1D chaotic encryption on the input image.

After performing 1D chaotic encryption the obtained image will be in RGB color space, we have to convert that image into YCbCr color space for further processing.

Fig.5.1.2 Encrypted Image

This is the YCbCr image obtained after conversion from RGB color space to YCbCr color space.

After conversion into YCbCr color space we need to extract the Y component alone. We will divide this YCbCr color space into two channels.

- Y-channel.
- CbCr-channel.

The Y-Component is extracted from the YCbCr image.

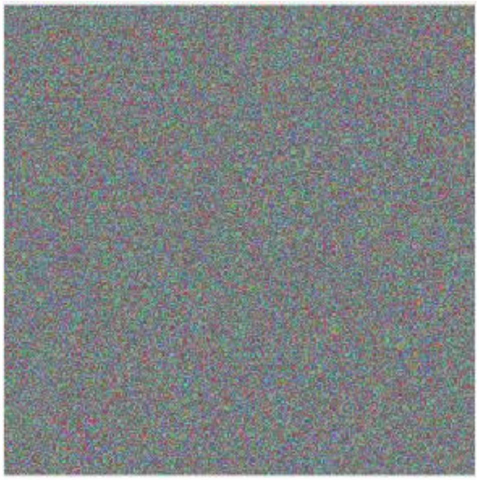
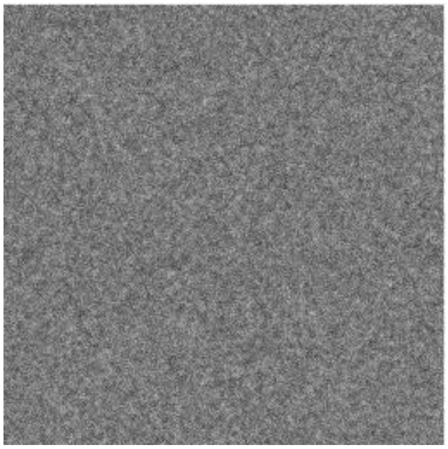


Fig.5.1.3 YCbCr Image



This the Extracted Y component image from YCbCr image.

After extracting Y component from YCbCr image we need to divide this Y Component into 8*8 non overlapping blocks.

Fig.5.1.4 Extracted Y component

After Converting the Y component image into 8*8 non overlapping blocks we need to perform DCT and quantization for the image which helps in image compression

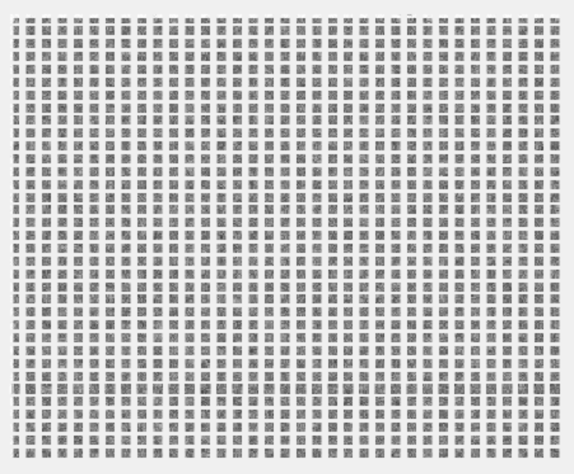
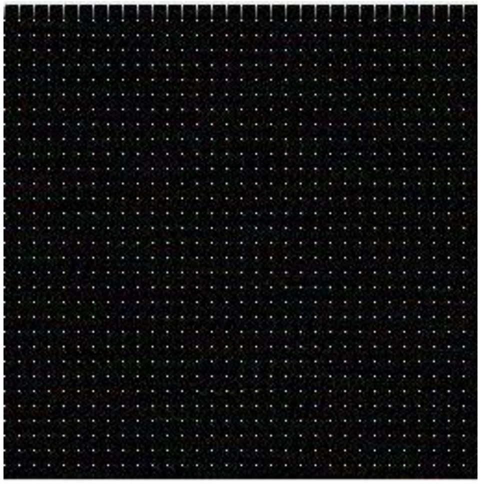


Fig.5.1.5 Y component Image into 8*8 non overlapping blocks



This is the image generated after applying DCT for each block which is generated by dividing the Y component into 8*8 non overlapping blocks. This image with white dots is generated represents the gap between each non-overlapping block. After applying Discrete Cosine Transform, the image looks black, as the output coefficeints are low in value and randomness of pixel increases.

Fig.5.1.6 After applying DCT for each block

This image is obtained after the quantization process. From this we can infer that, it is black in color because based on the quantization table, the pixel value is divided, that means the larger value is mapped to a very smaller value i.e., small floating values. As the value is nearer to 0, it resembles with black color.

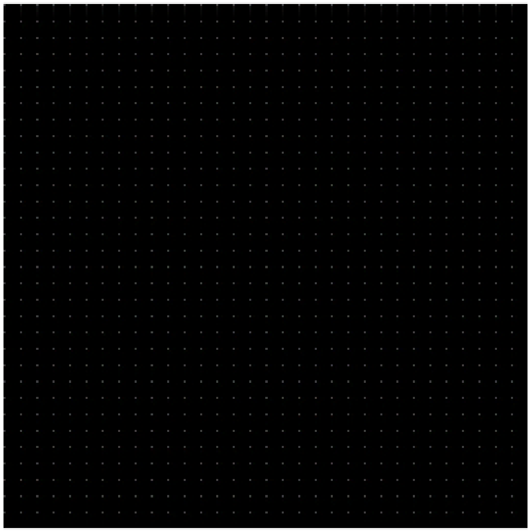


Fig.5.1.7 After applying Quantization to DCT image

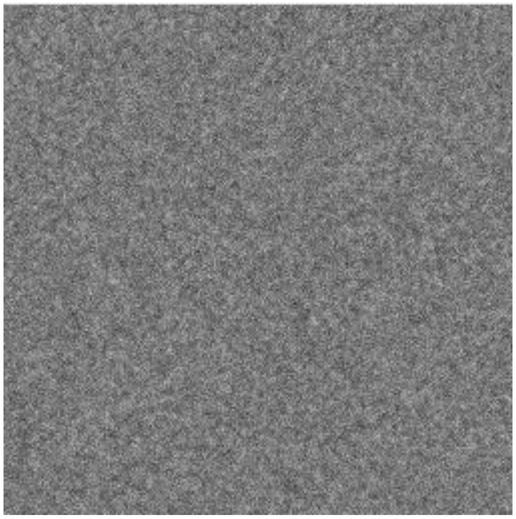


Fig 5.1.8 – Stego-Encrypted Image

This image is the stego-encrypted image. Here, the data stored is “Saaketh and Venkat”. This string has ASCII value. This ASCII is converted to 7-bit binary data and this data is encoded using run-length encoding and now, this data is embedded into image by clearing 3 bits from each pixel of the image and replacing with encoded text.

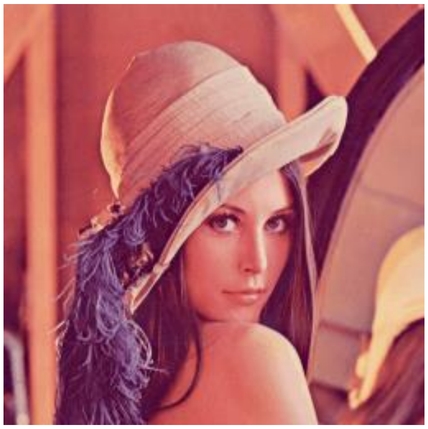


Fig 5.2.1 – Input Image (Lena)

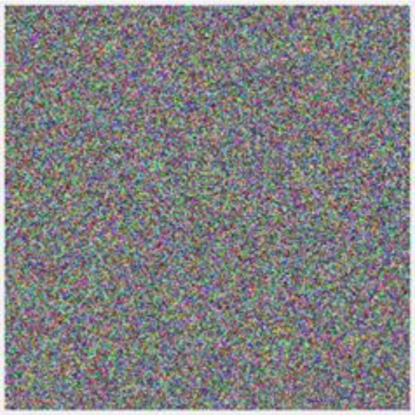


Fig 5.2.2 – Encrypted Image

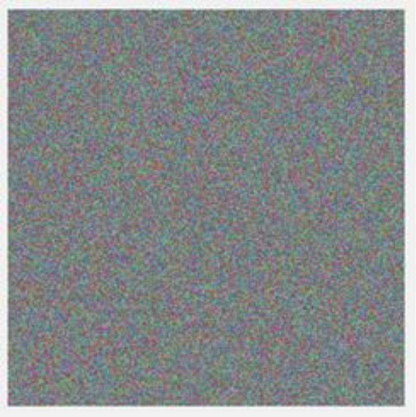


Fig 5.2.3 – YCbCr Image

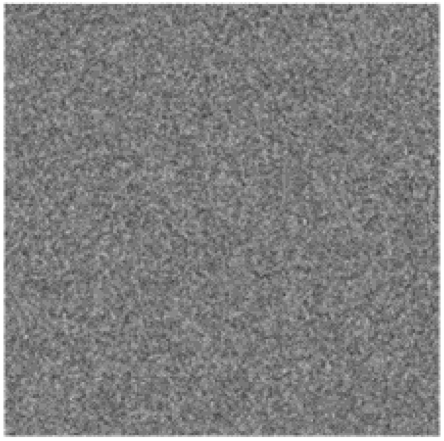


Fig 5.2.4 – Y Component Image

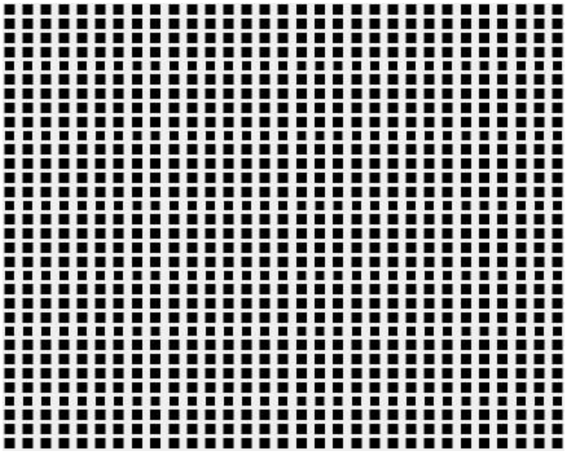


Fig 5.2.5 – Divided into 8*8 blocks

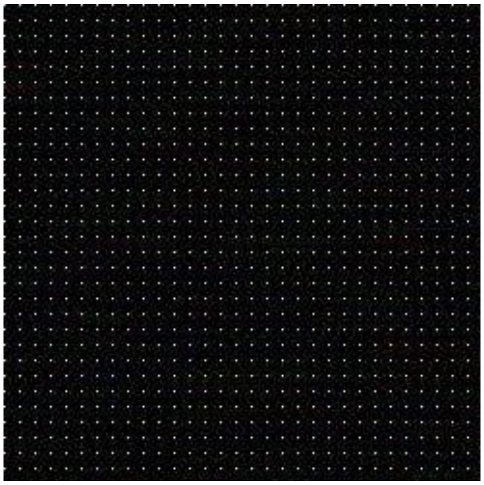


Fig 5.2.6 – After Applying DCT

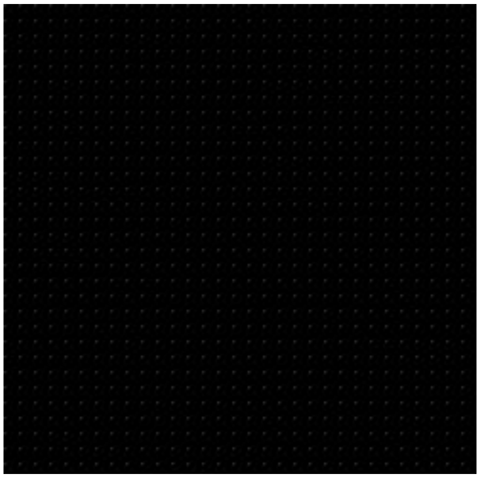


Fig 5.2.7 – After Applying Quantization

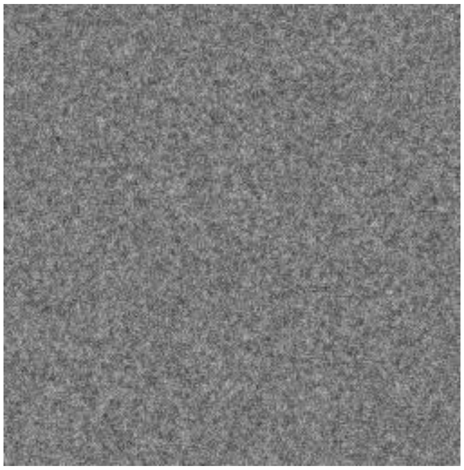


Fig 5.2.8 – Stego-Encrypted Img

CHAPTER 6

CONCLUSION

In the domain of Information security, the data need to be stored by applying various encryption techniques which makes the hacker to decrypt the image and text embedded in it. In conclusion, this project completely focused on the image encryption and data embedding into it which has a wide application in the field of medical and transportation and many more. It uses 1D chaotic map with the mixing of both confusion and diffusion process. Then from the encrypted Image Y-component image is extracted and later on the DCT and Quantization processes are done for distortion of pixels and reducing values of pixels by which security can be increased in the image. On the other hand, run-length encoding process is used for encoding the input string, which increases the security in various aspects.

Finally, to store the bits of data i.e., the secret message, each pixel is used. Usually, each pixel is of 8 bit binary value. Clearing 3 LSB bits and replacing the binary encoded data by splitting encoded data into bits of length 3. SO total 1,96,608 bits of data can be stored inside a 256 * 256 sized colour image. That means each image has an embedding capacity of 37.5% of its original image. Finally, a stego-encrypted image is generated which is secure and has a good embedding capacity.

24/4/24

Signature of the Guide: Student Register No: 124160012

124160081

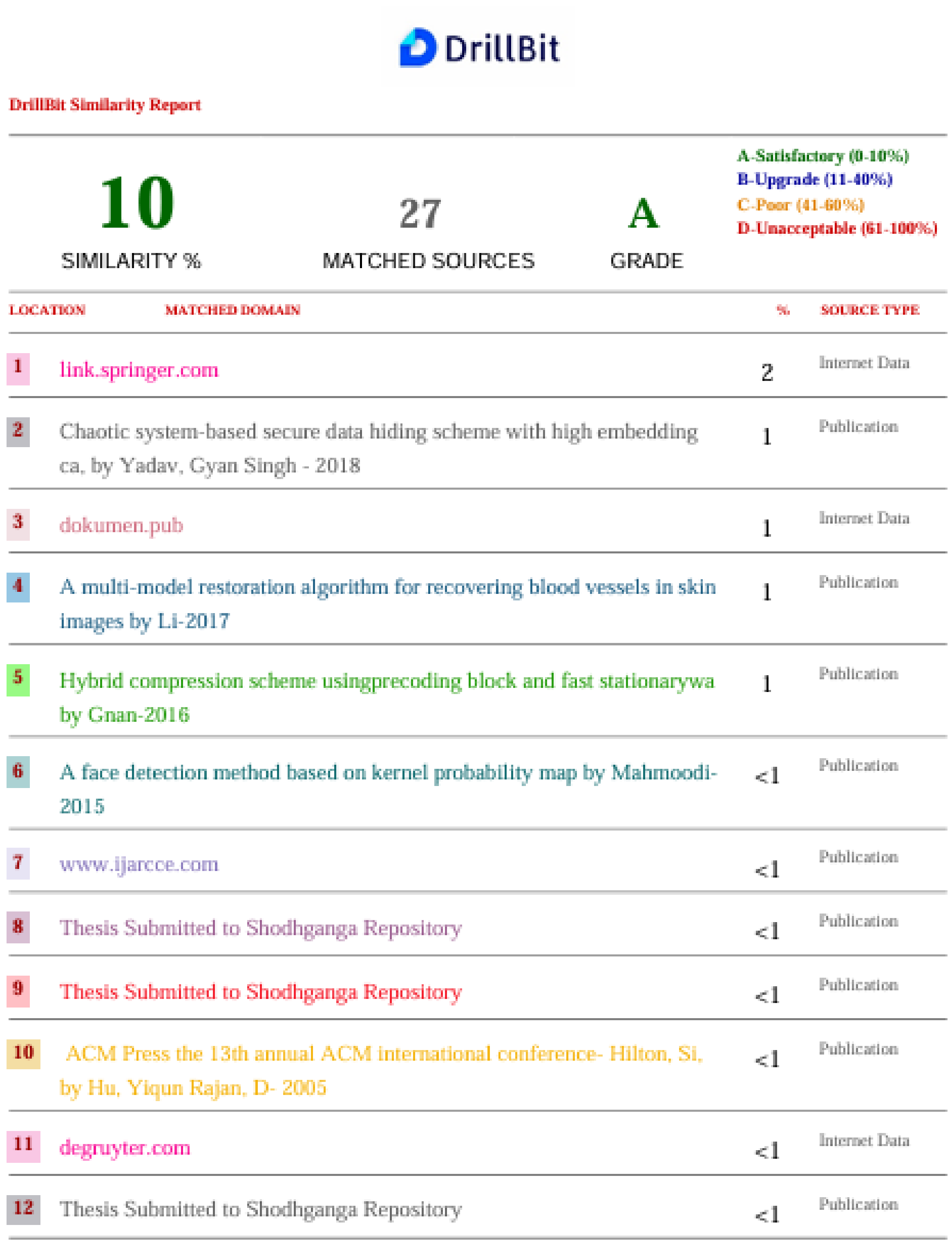
Name of the Guide: **Dr. Lakshmi .C,** AP-III, SEEE Name: Chakka Sai Venkat

Shastrula Saaketh Sharma

CHAPTER 7

REFERENCES

1. *CESCAL: A joint compression-encryption scheme based on convolutional autoencoder and logistic map* Sabha, I., Parah, S. A., Sarosh, P., & Islam, M. O. U. (2023). CESCAL: A joint compression-encryption scheme based on convolutional autoencoder and logistic map. Multimedia Tools and Applications, 1-30.
2. *A color image encryption technique using block scrambling and chaos* Hosny, K. M., Kamal, S. T., & Darwish, M. M. (2022). A color image encryption technique using block scrambling and chaos. Multimedia Tools and Applications, 81(1), 505-525.
3. *Improved data hiding method for securing color images(Base Paper)* Abdel-Aziz, Mostafa M., Khalid M. Hosny, and Nabil A. Lashin. "Improved data hiding method for securing color images." Multimedia Tools and Applications 80.8 (2021): 12641-12670.
4. *Color Image Encryption Technique Based on Chaos* Elshamy, A. M., Hussein, A. I., Hamed, H. F., Abdelghany, M. A., & Kelash, H. M. (2019). Color image encryption technique based on chaos. Procedia Computer Science, 163, 49-53.
5. *Autoencoder-based joint image compression and encryption* Wang, B., & Lo, K. T. (2024). Autoencoder-based joint image compression and encryption. Journal of Information Security and Applications, 80, 103680.
6. *An efficient chaos-based image compression and encryption scheme using block compressive sensing and elementary cellular automata* Chai, X., Fu, X., Gan, Z., Zhang, Y., Lu, Y., & Chen, Y. (2020). An efficient chaos-based image compression and encryption scheme using block compressive sensing and elementary cellular automata. Neural Computing and Applications, 32, 4961-4988.
7. Bhardwaj, R., & Khanna, D. (2015, December). *Enhanced the security of image steganography through image encryption*. In 2015 Annual IEEE India Conference (INDICON) (pp. 1-4). IEEE.
8. Sridevi, D. R., Vijaya, P., & Rao, K. S. (2013*). Image steganography combined with cryptography*. Council for Innovative Research Peer Review Research Publishing System Journal: IJCT, 9(1).
9. Tang, H., Jin, G., Wu, C., & Song, P. (2009, December). *A new image encryption and steganography scheme*. In 2009 international conference on computer and communications security (pp. 60-63). IEEE.
10. Sravani, S., & Raniith, R. (2021, July). *Image steganography for confidential data communication*. In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 01-05). IEEE.
11. *Combined image encryption and steganography technique for enhanced security using multiple chaotic maps* Aparna, H., & Madhumitha, J. (2023). Combined image encryption and steganography technique for enhanced security using multiple chaotic maps. Computers and Electrical Engineering, 110, 108824.



DrillBit

DrillBit Similarity Report

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

10	27	A
SIMILARITY %	MATCHED SOURCES	GRADE

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	link.springer.com	2	Internet Data
2	Chaotic system-based secure data hiding scheme with high embedding ca, by Yadav, Gyan Singh - 2018	1	Publication
3	dokumen.pub	1	Internet Data
4	A multi-model restoration algorithm for recovering blood vessels in skin images by Li-2017	1	Publication
5	Hybrid compression scheme usingprecoding block and fast stationarywa by Gnan-2016	1	Publication
6	A face detection method based on kernel probability map by Mahmoodi-2015	<1	Publication
7	www.ijarcce.com	<1	Publication
8	Thesis Submitted to Shodhganga Repository	<1	Publication
9	Thesis Submitted to Shodhganga Repository	<1	Publication
10	ACM Press the 13th annual ACM international conference- Hilton, Si, by Hu, Yiqun Rajan, D- 2005	<1	Publication
11	degruyter.com	<1	Internet Data
12	Thesis Submitted to Shodhganga Repository	<1	Publication