

Mini Project Report

on

Image Encryption using Visual Cryptography



By

Ishita Sinha (Reg. No.-202000506)

Shiv Shankar (Reg. No.-202000475)

Shweta Jha (Reg. No.-202000120)

Group Id – C3

*In partial fulfillment of requirements for the award of degree in
Bachelor of Technology in Computer Science and Engineering
(2023)*

Under the Project Guidance of

Prof. (Dr.) Sandeep Gurung

Associate Professor, Dept. of CSE

Sikkim Manipal Institute of Technology, Majitar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY

(A constituent college of Sikkim Manipal University)

MAJITAR, RANGPO, EAST SIKKIM – 737136

PROJECT COMPLETION CERTIFICATE

This is to certify that the below mentioned students of Sikkim Manipal Institute of Technology have worked under my supervision and guidance from **9th January 2023 to 29th April 2023** and successfully completed the Mini project entitled **“Image Encryption using Visual Cryptography”** in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Course
202000506	Ishita Sinha	B.Tech CSE
202000475	Shiv Shankar	B.Tech CSE
202000120	Shweta Jha	B.Tech CSE



Dr. Sandeep Gurung

Associate Professor

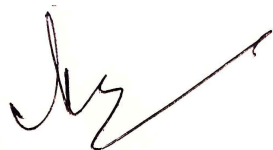
Department of Computer Science and Engineering

Sikkim Manipal institute of Technology

Majhitar, Sikkim – 737136

PROJECT REVIEW CERTIFICATE

This is to certify that the work recorded in this project report entitled “**Image Encryption using Visual Cryptography**” has been jointly carried out by **Ishita Sinha (Reg. 202000506)**, **Shiv Shankar (Reg. 202000475)** and **Shweta Jha (Reg. 202000120)** of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Mini Project Viva Examination.



Dr. Sandeep Gurung

Associate Professor

Department of Computer Science and Engineering

Sikkim Manipal Institute of Technology

Majhitar, Sikkim – 737136

CERTIFICATE OF ACCEPTANCE

This is to certify that the below mentioned students of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) have worked under the supervision of **Dr. Sandeep Gurung**, Assistant Professor, Department of Computer Science and Engineering from **9th January 2023 to 29th April 2023** on the project entitled **“Image Encryption using Visual Cryptography”**.

The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Project Venue
202000506	Ishita Sinha	SMIT
202000475	Shiv Shankar	SMIT
202000120	Shweta Jha	SMIT



Dr. Udit Kumar Chakraborty

Professor & Head of the Department

Computer Science & Engineering Department

Sikkim Manipal Institute of Technology

Majhitar, Sikkim – 737136

DECLARATION

We, the undersigned, hereby declare that the work recorded in this project report entitled **“Image Encryption using Visual Cryptography”** in partial fulfillment for the requirements of award of B.Tech (CSE) from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bonafide project work carried out at **“SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY”** under the supervision and guidance of **Dr Sandeep Gurung**, Assistant Professor, Department of Computer Science and Engineering.

The results of this investigation reported in this project have so far not been reported for any other Degree or any other Technical forum.

The assistance and help received during the course of the investigation have been duly acknowledged.



Ishita Sinha (Reg. No.-202000506)



Shiv Shankar (Reg. No.-202000475)



Shweta Jha (Reg. No.-202000120)

ACKNOWLEDGMENT

We take this opportunity to acknowledge indebtedness and a deep sense of gratitude to our guide **Prof. (Dr.) Sandeep Gurung** for his/her valuable guidance and supervision throughout the course which shaped the present work as it shows.

We pay our deep sense of gratitude to **Prof. (Dr.) Udit Kumar Chakraborty, HOD, Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us the opportunity to work on this project and providing all support required.

We are obliged to our project coordinators **Dr. Sandeep Gurung** and **Mr. Biraj Upadhyaya** for elevating, inspiring and supervising the completion of our project.

We would also like to thank any other staff of the **Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us continuous support and guidance that has helped us in completion of our project.



Ishita Sinha (Reg. No.-202000506)



Shiv Shankar (Reg. No.-202000475)



Shweta Jha (Reg. No.-202000120)

DOCUMENT CONTROL SHEET

1	Report No	CSE/Mini Project/Internal/B.Tech/Section/C3/2023
2	Title of the Report	Image Encryption using Visual Cryptography
3	Type of Report	Technical
4	Author	Ishita Sinha Shiv Shankar Shweta Jha
5	Organizing Unit	Sikkim Manipal Institute of Technology
6	Language of the Document	English
7	Abstract	An attempt has been made to explore the integration of Visual Cryptography and Convolution Neural Networks to improve image encryption. An approach is being made to propose a method that uses random grids to generate shares and a CNN to decode them, providing enhanced security against attacks. The system and its performance is evaluated using metrics such as accuracy, speed, and robustness against various attacks. Results show that the proposed method provides better security and efficiency than traditional Visual Cryptography schemes.
8	Security Classification	General

9	Distribution Statement	General
---	------------------------	---------

TABLE OF CONTENTS

Chapter		Title	Page No.
		Abstract	
1		Introduction	1 - 7
2		Literature Survey	8 - 13
3		Problem Definition	14
4		Solution Strategy	15 - 16
5		Design	
	5.1	Block Diagram for Image Encryption Using Visual Cryptography	17
	5.2	Multiple secret sharing and information hiding on square random grids	18
	5.3	Visual Cryptography using Convolution Neural Networks	19

6		Implementation details	
	6.1	Implementation of Visual Secret Sharing using two random grids	20 - 21
	6.2	Implementation of multiple information hiding algorithm on square	22 - 24
	6.3	Implementation of multiple information hiding algorithm on same pair of rectangular random grids	24 - 26
	6.4	Implementation of multiple information hiding algorithm on circular cipher grids vii	26 - 28
	6.5	Using Machine Learning to enhance Visual Cryptography using Random Grids	28 - 32
7		Results and discussion	
	7.1	Performance Metrics	33
	7.2	Results and Analysis	33 - 38
8		Conclusion	39
9		Limitations and Future Scope of the project	40
10		Gantt Chart	41

11		References	42 - 43
12		Plagiarism Report	44

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.1	Naor and Shamir's 2 out of 2 Algorithm	2
2.1	A 2 x 2 array with 4 subpixels	8
2.2	4 out of 4 visual secret sharing problems	9

4.1	Encryption using random-grids	15
5.1	Flow diagram for Image Encryption	17
5.2	Multiple secret sharing and information hiding on square random grids	18
5.3	Visual Cryptography using CNN	19
6.1	The flowchart of Random grid visual secret sharing scheme	20
6.2	Flowchart for multiple information hiding algorithm on same pair of rectangular random grids	25
6.3	Flowchart for multiple information hiding algorithm on circular cipher grids	27
ix		
6.4	Encoding the Secret image to R1 and R2	29
6.5	High Level Architecture of the Training Model	29
6.6	Decoding of output (predicted and test)	30
6.7	Summary of the Sequential Model used	31

7.1	Visualization of the training and validation loss over the epochs.	38
7.2	Visualization of the training and validation accuracy over the epochs	38
10.1	Gantt chart	41

x

LIST OF TABLES

Table No.	Title Name	Page no.
2.1	Literature Survey	8 - 13
7.1	Representation of input images, output images and PSNR for the 4 angles	34

7.2	Representation of input images, output images and PSNR for various angles	35
7.3	Representation of input images, random grids, output images and PSNR for the 4 angles	36
7.4	Comparison of Prediction vs Test Decoded images and their PSNR	37
7.5	Comparing the loss and accuracy for the initial and final Epoch	38

ABSTRACT

Visual cryptography is a secure communication technique that encrypts communications with visuals in such a way that decryption can be accomplished solely through sight reading. Secure digital transmission is offered by Visual Cryptography. It works by dividing a picture or piece of text into several shares such that when the shares are layered on top of one another, the original image or piece of text is seen.

An approach has been made to explore the integration of Visual Cryptography and Convolution Neural Networks to improve image encryption. An attempt is being made for a method that uses

random grids to generate shares and a CNN to decode them, providing enhanced security against attacks. The system and its performance is evaluated using metrics such as accuracy, speed, and robustness against various attacks. Results show that the proposed method provides better security and efficiency than traditional Visual Cryptography schemes.

1. INTRODUCTION

1.1. Introduction

Visual Cryptography (VC) is a technique that allows secret sharing of images or messages between multiple parties without revealing the secret to unauthorized users. It involves dividing the original image into multiple shares, such that the secret can only be revealed when the shares are stacked in a specific manner. Traditional VC schemes use simple techniques like XOR or AND operations to generate shares, which may not provide sufficient security against modern attacks.

This paper talks about image encryption using visual cryptography. Image encryption is a procedure which converts a plain image to an encrypted image by employing a secret key.

Visual Cryptography (VC) is a cryptographic scheme which can decode concealed images without any cryptographic computations. [1]

To address this issue, we propose a novel VC scheme that combines random grids and Convolutional Neural Networks (CNNs) to generate and decode shares, respectively. In this scheme, a random grid is superimposed on the original image to generate two shares. These shares are then sent to the CNN, which is trained to decode them and reconstruct the original image. The CNN-based decoding process provides enhanced security against attacks by incorporating non-linear transformations and advanced feature extraction techniques.

The proposed scheme can find applications in secure image sharing, digital watermarking, and authentication systems.

1.2. Secret Sharing Scheme : (k:n scheme)

The basic model consists of a printed page of ciphertext (which can be sent by mail or faxed) and a printed transparency (which serves as a secret key). [1] The original text is revealed by placing the key over the ciphertext.

Naor and Shamir [1] on visual cryptography provided the most straightforward VC algorithm. For each image that was encrypted, two shares would be generated ($n = 2$), and in order to decrypt an image, these two shares would need to be superimposed ($k = 2$). The 2 out of 2 algorithm functions by substituting two pixels for each original image pixel at the most fundamental level. Every pixel in the original image is read; if a white pixel is found, one of the first two rows in Figure 1 is chosen at random, and each share is given a 2 pixel block as illustrated in the third and fourth columns.

Similar to this, if a black pixel is found, one of the final two rows is picked at random with an equal chance, and a subpixel is then assigned to each share from that row. [5]

When two shares are superimposed, the resulting pixel will be white if two white pixels overlap. In contrast, a black pixel in one share will result in a black pixel if it coincides with a white or black pixel in the other share. This suggests that the shares are superimposed to represent the Boolean OR function. [5]

Pixel	Probability	Share1	Share2	Share1 Share2
	50%			
	50%			
	50%			
	50%			

Figure 1.1. Naor and Shamir's 2 out of 2 Algorithm [1]

The necessity to represent one pixel by many pixels in each share makes this technology, which is good at providing security, susceptible to pixel expansion. As a result, the photos must be encrypted and decrypted more frequently. The decoded image is bigger than the original as well. Hence, another method of the Visual Secret Sharing scheme is used, called Random Grids.

The issue of exchanging two secret images in two shares in visual cryptography may have been addressed for the first time by Wu and Chen[1]. In two seemingly unrelated shares, let's say S1 and S2, they hid two secret binary images.

- The first secret is obtained by stacking the two shares: $S1 \otimes S2$

‘ $\otimes$ ’ represents superimposition, including the OR or XOR operation

- The second share: $S1^{\wedge\Theta} \otimes S2$

‘ $S1^{\wedge\Theta}$ ’ represents the resulting rotation of $S1$ through an angle Θ

One needs to possess lots of shared images for different secret images during transmissions. This property results in inefficiency, several efforts to hide more secret images into two shared images were being made. Unfortunately, the existing visual secret sharing schemes restrict the number of secret images.

1.2.1. On square random grids

- **Random grid generation (first share, $r1$)**

Random grid consists of a transparency which comprises transparent and opaque pixels arranged randomly and is designed in such a way that when being superimposed, it reveals the secret to the Human Visual System (HVS) without any computational parameters. It is a two-dimensional array of pixels and every pixel is either transparent or opaque. Transmission of light through these chosen pixels is random. Opaque pixels block out light whereas transparent pixels allow light to pass through. Number of opaque pixels (denoted by O), $P(O) = 1/2$ and the number of transparent pixels (denoted by Tr), $P(Tr) = 1/2$.

Thus the average light transmission of a random grid is also $1/2$. If we assume ‘ R ’ to be the random grid then $T(R) = 1/2$. For a certain pixel ‘ r ’ in random grid R the probability of r to be transparent is equal to that of r being opaque therefore:

$P(r=0) = P(r=1) = 1/2$; where 0 denotes opaque and 1 denotes transparent.

Function to generate the random grid: `np.random.randint(2, size=(l,b))`.

- **Binarization (binarized image, secret)**

Image binarization is the process of taking a grayscale image and converting it to black-and-white. Hence reducing the information contained within the image from 256 shades of gray to 2 that is black and white, a binary image. This is called image thresholding, although thresholding may produce images with more than 2 levels of gray. It is a form of segmentation, whereby an image is divided into constituent objects. This is a task performed to extract an object from an image.

Approach:

```
def binarize(img,l,b):  
    if(img[i,j] > 128 and img[i,j]<255), set img[i,j]=1  
    else, set img[i,j]=0
```

- **Generation of second random matrix (resulting from first share and the binarized image, r2)**

Each pixel value of the binarized image is compared with the 0 bit and the values for the second secret matrix are set accordingly.

Approach:

Rotated image through 0 degree:

```
def create_r2_at_0(secret,r1,r2,l,b):  
    if(secret[i,j]==0), for  $0 \leq i \leq l$  and  $0 \leq j \leq b$ , set  $r2[i,j] = r1[i,j]$   
    else, set  $r2[i,j] = 1 - r1[i,j]$ 
```

Rotation through 90 degree :

```
def create_r2_at_90(secret,r1,r2,l,b):  
    if(secret[j,l-i-1]==0), set  $r2[j,l-i-1] = r1[i,j]$   
    else, set  $r2[j,l-i-1] = 1 - r1[i,j]$ 
```

Rotation through 180 degree:

```
def create_r2_at_180(secret,r1,r2,l,b)
    if(secret[l-i-1,l-j-1]==0), set r2[l-i-1,l-j-1] = r1[l-j-1,i]
    else, set r2[l-i-1,l-j-1] = 1 - r1[l-j-1,i]
```

Rotation through 270 degree:

```
def create_r2_at_270(secret,r1,r2,l,b)
    if(secret[l-j-1,i]==0), set r2[l-j-1,i] = r1[j,l-i-1]
    else, set r2[l-j-1,i] = 1 - r1[j,l-i-1]
```

• Superimposition of the two generated shares (XOR operation)**Approach:**

Check if((r1[i,j]==1 and r2[i,j]==0)or(r1[i,j]==0 and r2[i,j]==1)), set final[i,j] = 1
 Else, set final[i,j] = 0

Repeat the above for all the other rotated images.

1.2.2. On the same pair of rectangular random grids

Three images, each with 1/3rd width of a rectangular grid and height as 200 are taken such that total width of the very grid is obtained after merging them together. The first random grid (r1) is obtained with the width as 120*m and height as 200, where 'm' being the total number of images taken. Now, the second random grid (r2) is generated by using image input image and the first random grid. This very approach can also be used for the implementation of multiple secret sharing using circular grids. After getting the two random grids as an input, superimposition is performed to get the final encrypted image, accordingly the value of a random number 'k' taken which varies from 1 to m.

1.2.3. On circular grid

The main advantage of this scheme is that it does not use the basic matrices to encrypt the shares and so the problem of pixel expansion is completely eliminated. As a result the sizes of the

original image as well as the encrypted shares are the same which ultimately reduces the overhead of transmission, communication and storage. The problem with random grids is that it is only possible to encrypt a single image.

The two different hidden images can be obtained by stacking the shares and then rotating the grids. Requirement is for both the grids as well as the knowledge about the angle of rotation for which the images would be obtained.

1.3. Visual Cryptography using Convolution Neural Network (CNN)

Visual cryptography makes an extensive use of Convolutional neural networks (CNN) in computer vision tasks. It is a method of encrypting data that enables many individuals to secretly share an image or message. The security and sturdiness of visual cryptography have been improved by the usage of CNN. In order to produce shares or layers more securely and effectively, CNN can be trained to learn intricate patterns and characteristics from the input images. The visual clarity and authenticity of the rebuilt image can also be enhanced using CNNs. In visual cryptography, CNNs' capacity to handle vast and complicated images is one of their key advantages. Without the requirement for human feature extraction, CNNs can process high-resolution photos and learn the underlying patterns and characteristics in the image. The security and robustness of the approach can be determined through the complexity and randomness of the shares. CNNs can handle huge and complicated images, learn intricate patterns and characteristics from input images, and enhance the security and robustness of the technique, making it a crucial tool in visual cryptography.

1.3.1. Image classification

Image classification is the process of assigning an image as input to a model created using a particular technique, which then outputs the class or the probability that the image belongs to that class. Supervised learning refers to the procedure through which we assign a specific class to a given image.

- **Algorithm**

Step 1: Load the pre-trained model: `tf.keras.applications.MobileNetV2()`

Step 2: Load and pre-process the input image:

- `img_path='image_file'`
- `img=cv2.imread(img_path)`
- `img=cv2.resize(img,(224,224))`
- `tf.keras.applications.mobilenet_preprocess_input(img)`

Step 3: Make a prediction on the input image: `model.predict(np.expand_dims(img,axis=0))`

Step 4: Print the predicted class label:

- `predicted_class=tf.keras.applications.mobilenet_v2.decode_predictions(prediction, top=1) [0][0][1]`
- `print ('Predicted class:',predicted_class)`

2. LITERATURE SURVEY

2.1 Paper Details: Visual Cryptography, 1995

Author: Moni Naor and Adi Shamir

A novel form of cryptographic technique was taken into consideration by Moni Naor and Adi Shamir [1], which may decode hidden images without the use of cryptographic computations. The system is really simple to use and completely secure. We expand it into a visual version of the k out of n secret sharing dilemma where a dealer gives a transparency to each of the n users. Any k of them can see the image by stacking their transparency, but any $k-1$ of them gain no knowledge about it.

In this article, we look at the issue of creating a completely safe kind of visual decoding that may be used to encrypt written content (printed text, handwritten notes, photos, etc.). The fundamental model consists of a printed page containing ciphertext and a printed transparency that may be mailed or faxed (which serves as a secret key).

Even though each one is identical to random noise, the original cleartext can be read by placing the transparency with the key over the page with the ciphertext. In that each page of ciphertext is decrypted using a distinct transparency, the system is analogous to a one-time pad.

Even though each one is identical to random noise, the original cleartext can be read by placing the transparency with the key over the page with the ciphertext. In that each page of ciphertext is decrypted using a distinct transparency, the system is analogous to a one-time pad.

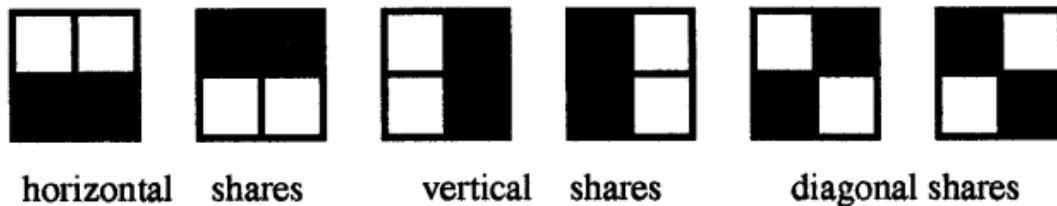


Figure 2.1: A 2×2 array with 4 subpixels [1]

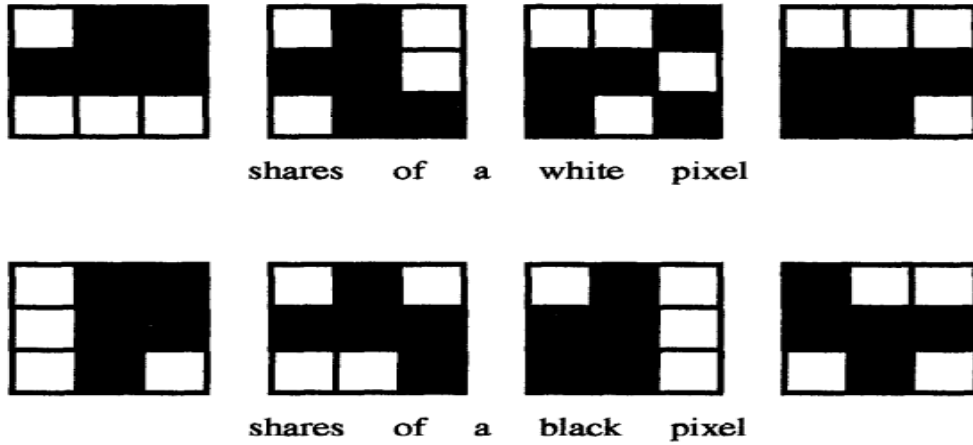


Figure 2.2: 4 out of 4 visual secret sharing problem [1]

A general 'k' out of 'k' scheme: We now outline two general architectures that, respectively, use 2^k and 2^k-1 subpixels to address any k out of k visual secret sharing problems.

Theorem : In any k out of n scheme, $\alpha \leq 1/(2^k-1)$ and $m \geq 2^k-1$. Here, ' α ' represents the relative difference in weight between combined shares from the white and black pixels and it should be as large as possible while 'm' represents the number of pixels in a share and it should be as small as possible.

To overcome the issues related with above mentioned techniques, we got to know about a more generalized scheme called k out of n strategy.

A common k-out-of-n strategy [1]: In this part, a k out of n scheme is built. The transition from a k out of k scheme to a k out of n scheme is what we demonstrate.

Theorem : For every n and k, a visual secret sharing system exists with the following parameters: $m = n^k \cdot 2^k - 1$, $\alpha = (2e)^{-k} \sqrt{2\pi k}$ and $r = n^k(2^k-1!)$.

2.2. Paper Details: Encryption of pictures and shapes by random grids

Author: O. Kafri and E. Keren

O. Kafri and E. Keren [2] introduced a method of encrypting pictures and shapes by random grids. They defined a random grid as a matrix of pixels in which each pixel can be fully transparent or totally opaque. The choice between the alternatives is made by a coin-flip procedure. The average transmission of the grid is 1/2, and there is no correlation between the values of different pixels in the array.

They also introduced three algorithms of generating such random grids.

In the first method, pictures in regions of gray level 1 are copied exactly from the corresponding regions of a specific random grid. Regions of gray level 0 are chosen as the complement of the master grid, pixel by pixel.

In the second method, Regions of gray level 1 are copied from the master as before. The pixels in regions of gray level 0 are chosen randomly, independent of the key.

In the third method, Pixels in regions of gray level 1 are chosen at random, and pixels with gray level 0 are complementary to those of the master grid.

They found that the result of algorithm 1 had the highest contrast.

An extension of this method was to combine the three algorithms to encode a picture with three gray levels. Pixels with gray levels 1, $1/2$, and 0 are chosen according to the master grid, a random value, and the negative of the master, respectively. The method can be extended to the encryption of pictures with more than three gray levels.

2.3. Paper Details: Multi-image encryption by circular random grids

Author: Tzung-Her Chen*, Kuang-Che Li

Tzung-Her Chen* and Kuang-Che Li [4] puts forth an unique RG-based VSS scheme that encodes multiple secret images. The method encrypts a number of secret images into two circular cipher-grids, then decrypts the images by stacking the two circular cipher-grids to obtain the first secret and gradually rotating one circular cipher-grid at a fixed degree (based on the number of the secret images encrypted) to reveal additional secrets. The suggested technique offers several advantages over traditional VC-based VSS, including a better capacity for secret sharing, the absence of pixel expansion, and the absence of the need to redesign a complicated VC codebook.

Table 2.1: Literature Survey

Sl. No.	Author	Paper and Publication Details	Findings	Relevance
1	Moni Naor and Adi Shamir	Paper: Visual Cryptography Publication Details: Proceedings of advances in cryptology: Eurocrypt94, lecture notes in computer science, vol.950, pp.1-12,1995	• Learnt basic image encryption techniques and proposed schemes: a.General k out of k scheme b.General k out of n scheme: The proposed scheme is perfectly secure however it suffers from pixel expansion resulting in loss of contrast. •Codebooks are difficult. •The practical implementation of a k out of n visual scheme even for small values of k and n.	The paper helped us in understanding the approach being used in image encryption and decryption based on several schemes which can decode the concealed images without any cryptographic computations. We analyzed that there can be many possible enhancements and extensions of the basic model introduced in the paper.

			•Full transparency is being provided to each of the users.	
2	O. Kafri and E. Keren	Paper: Encryption of pictures and shapes by random grids Publication Details: Optics letters 12.6 (1987): 377-379	•Benefits of using Random Grids for encryption algorithms for generating different types of random grids. •Finding the best algorithm for a high PSNR ratio. We found that algorithm 1 had the best output.	This paper helps us understand how the various algorithms of random grids can be beneficial as there is no pixel expansion. Also selected the best algorithmic approach for Visual Cryptography
3	Sandeep Gurung et. al.	Paper: Multiple information hiding using cubical approach on random grids Publication Details: International Journal of Computer Network and Information Security 7.11 (2015): 54-63.	•Learned the implementation of multiple information hiding using square random grids. •Learned about the usage of PSNR and Contrast to determine the quality of superimposed images.	This paper helps us understand how to implement multiple information hiding by rotating the random grids.
4	Tzung-Her	Paper: Multi-image encryption by circular random grids	•This paper talks about the overview of Visual secret sharing (VSS) and	This paper helps us understand the benefits of

	Chen and Kuang- Che Li	Publication Details: Information Sciences 189 (2012): 255-265.	its random grid based scheme which has been used to encrypt multiple secret images into two circular cipher-images. This approach is only restricted to up-to four secret images. •In order to retrieve the secret image, the cipher-grids are layered together. One circular cypher grid is then gradually rotated to reveal other secrets.	random-grids over visual cryptography and how we can encrypt multiple data into circular cypher-grids.
5	Rikiya Yamashi ta & Mizuho Nishio & Richard Kinh Gian Do & Kaori Togashi	Paper: Convolutional neural networks: an overview and application in radiology Publication Details: The Author(s) 2018	This paper talks about Convolutional neural networks (CNN), its multiple building blocks, such as convolution layers, pooling layers, and fully connected layers. It also talks about some challenges while using CNN like small dataset, overfitting and the	This paper helps us understand the basics of CNN and how we can implement an efficient CNN model with Visual Cryptography.

			techniques to minimize them.	
--	--	--	------------------------------	--

3. PROBLEM DEFINITION

- Digital images are used for communication in various areas now that technology has advanced. These pictures can include private and delicate information in them. These images are susceptible to a number of problems when transmitted over open networks, including modification and unauthorized access. Therefore, images must be safeguarded from various security intrusions.
- Visual cryptography has three main drawbacks: the large size, the need for the codebook to be redesigned for specific applications, and the ability to encode only one secret image at a time [4]. Another problem with visual cryptography is pixel expansion which involves a loss of contrast that degrades the reconstructed image's quality and has an impact on the image's aspect ratio.
- The random-grids (RGVSS) approach is effective, but the main drawback is that it necessitates standardizing the secret image and can only encrypt one secret image at a time. Comparing the superimposed (XOR) and secret image, the superimposed image has lesser psnr value.
- The conventional method could only encode one secret at a time. Wu and Chen [8] initiated the concept of multiple secret sharing in rectangular shares wherein one of the shares was rotated anticlockwise at various angles of 90° , 180° and 270° to recover multiple secret information. The predictable angle of orientation was a major drawback of the approach [3].

- Circular random-grids have the drawback that when more secret data is added, the visual quality of the decrypted image degrades. In the circular cypher-grid, the size of the sector-pixels differs. Convolutional Neural Network (CNN) with Visual Cryptography involves generating high quality shares for a secret image but is significantly slower due to an operation such as maxpool.

4. SOLUTION STRATEGY

The original image has to be standardized/thresholded into a binary format with only two levels of gray (**0/1**), level **0** for the lowest pixel value and level **1** for the maximum.

The random grid will be comprising of two dimensional array of pixels either fully transparent or fully opaque and the choice between the alternative is totally random. The average transmission of the grid is $\frac{1}{2}$.

Considering the aspect ratio of random grid and standardized image equal, pixels in regions of gray level 1 are copied exactly from the corresponding regions of the random grid, which is considered the master grid. Regions of gray level 0 are chosen as complement and the transmission in these regions are 0 when two grids are superimposed [2]. The resulting grid will be the encoded grid.

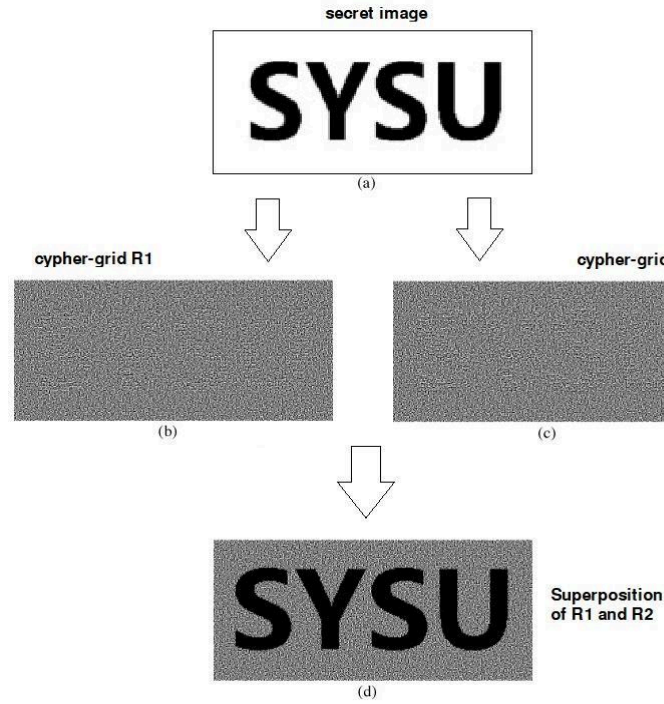


Figure 4.1. Encryption using random-grids

In a rectangular grid, three images are taken, each with a height of 200 and a width of $\frac{1}{3}$ of a rectangular grid, so that when they are combined, the whole width of the grid is acquired. With a width of $120 \times m$ and a height of 200, the first random grid (r1) is generated. Now, using the input image and the first random grid (r2) is generated.

In order to create the circular cypher-grid A, we overlay two circular pictures in white and black and then assign pixels of the 360 sector-blocks as transparent or opaque. For circular cypher-grid B, the RGVSS scheme must be implemented. The procedure for generating circular cipher-grids are shown below and in Fig. 7. Decryption undertakes the same process as conventional visual secret sharing by VC and RG. Secret messages are revealed after stacking two cipher-grids together if the angles of A and B are matched [4].

Convolutional Neural Network (CNN) is a type of Deep Learning Algorithm which is generally used for image classification and detection. The implementation of CNN model with Visual

Cryptography includes training and testing the model on (secret, r1 and r2) which later generates (y_pred) as the predicted r2.

After a few sets of training the CNN model, it can be used to generate high-quality r2 shares(y_pred) for new secret images.

5. DESIGN

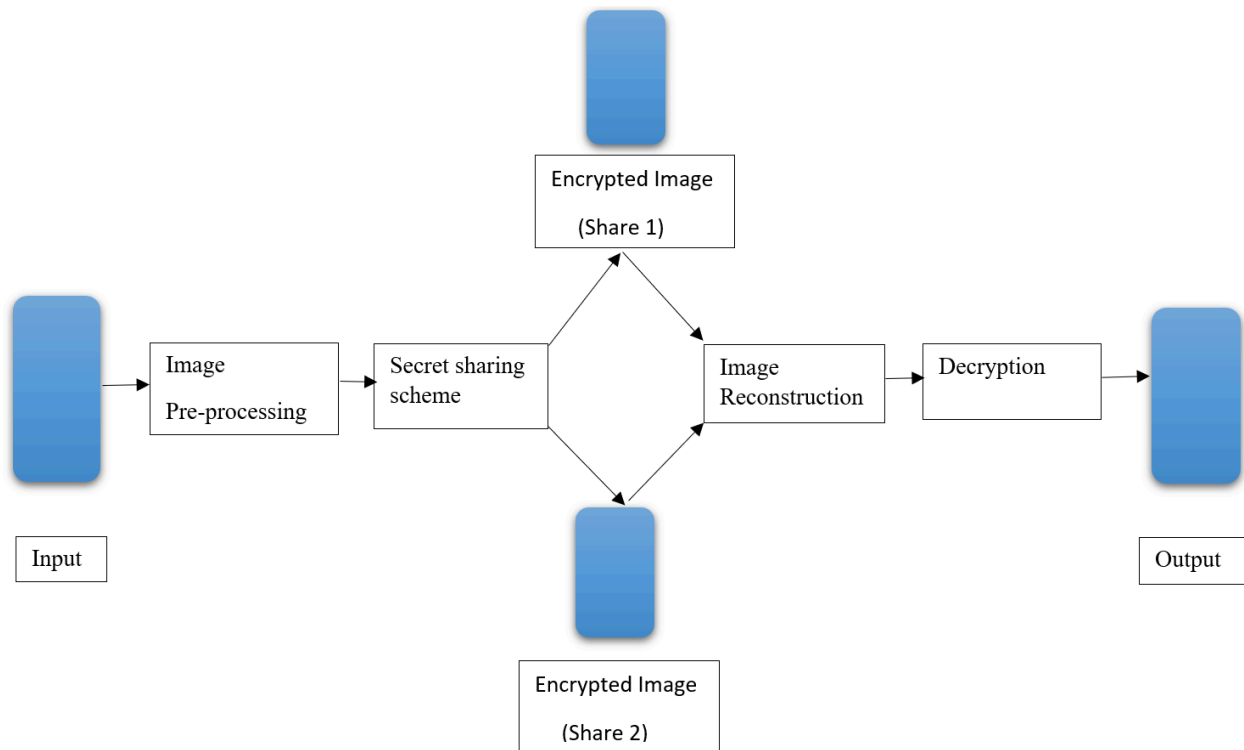


Figure 5.1. Flow diagram for Image Encryption

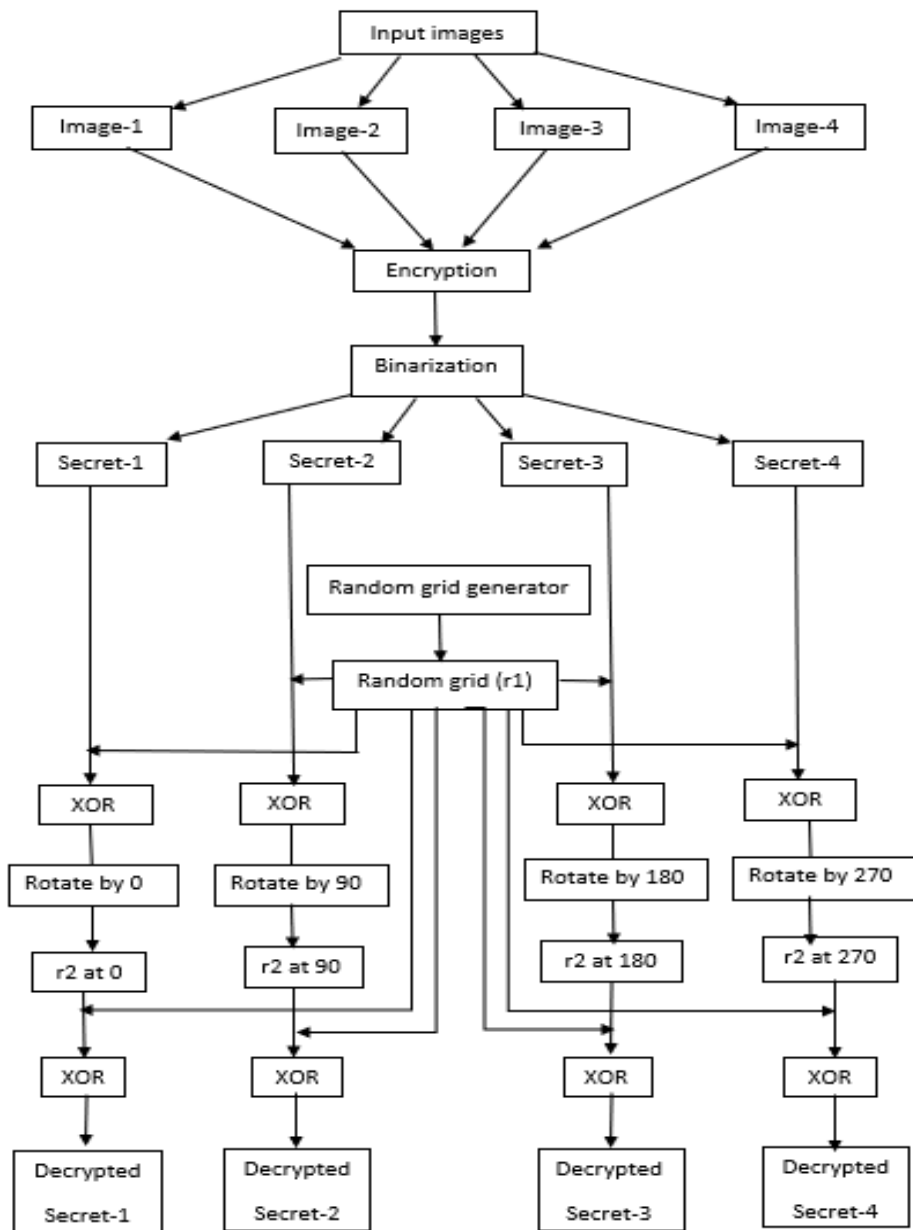


Figure 5.2. Multiple secret sharing and information hiding on square random grids

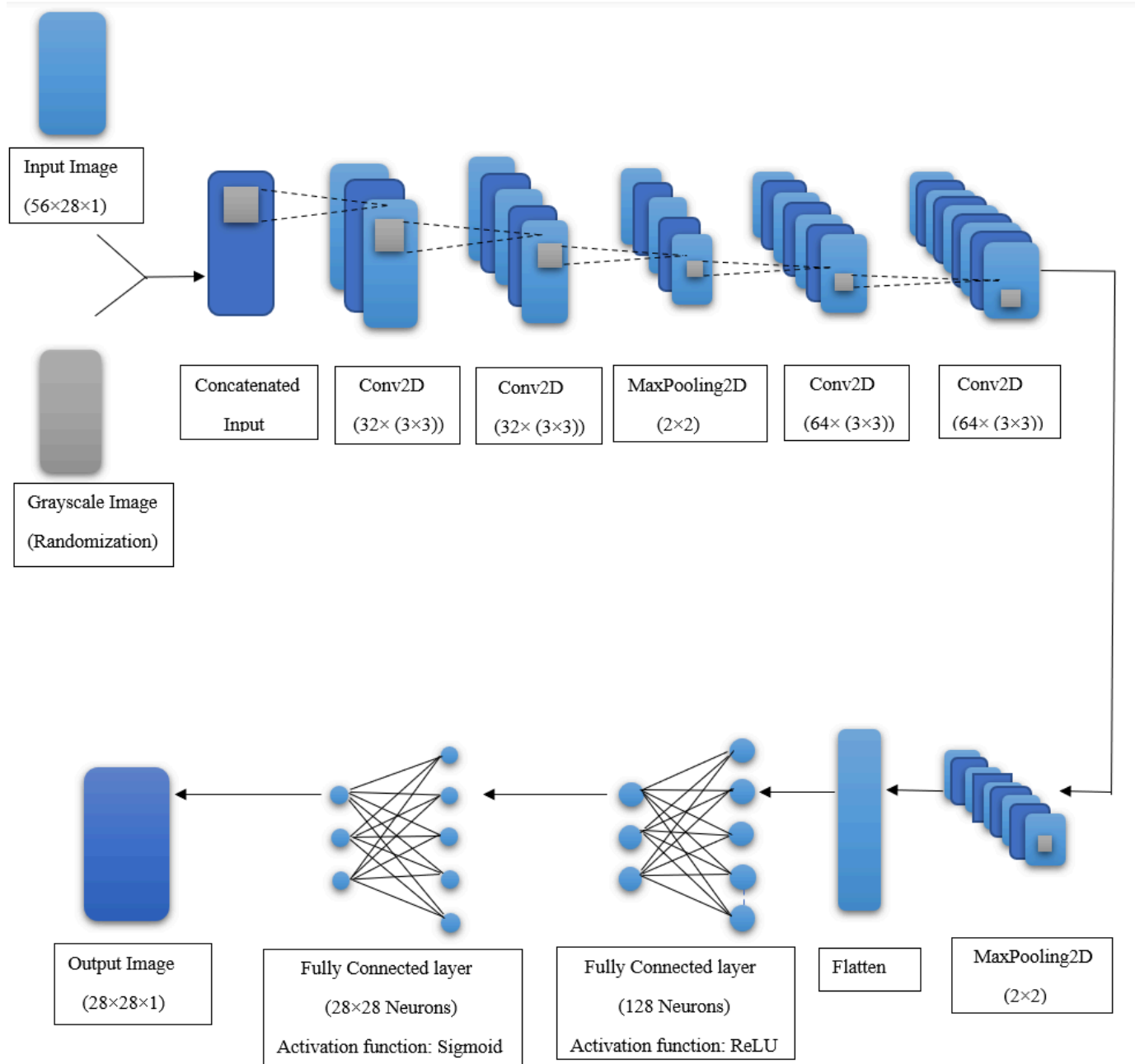


Figure 5.3. Visual Cryptography using CNN

6. IMPLEMENTATION DETAILS

6.1. Implementation of Visual Secret Sharing using two random grids

6.1.1. Introduction

This is the simplest implementation of the Visual Secret Sharing scheme discussed above. In this method, two random grids namely R1 and R2 are generated from the image S that needs to be encrypted. Then, R1 and R2 are superimposed using XOR operation to recover the image. This is the basic algorithm that is followed in all implementations in this report.

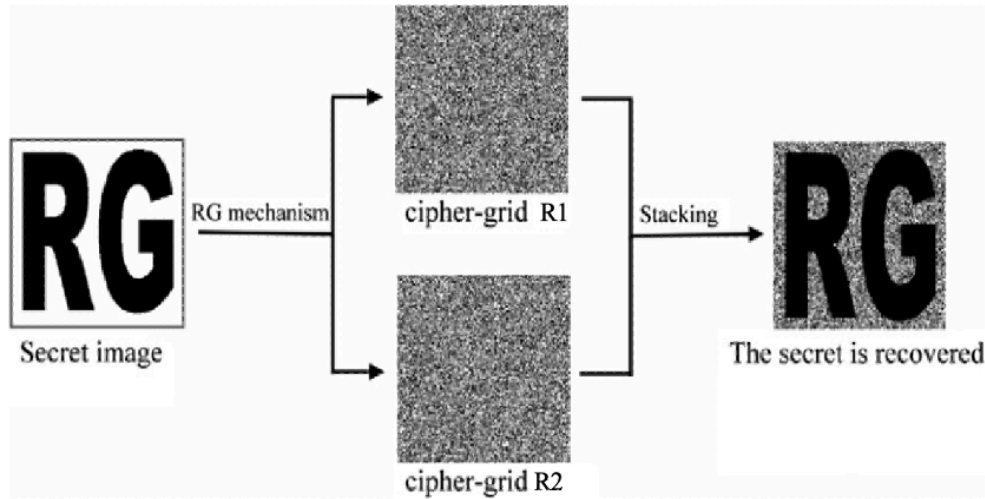


Figure 6.1. The flowchart of Random Grid Visual Secret Sharing scheme [4]

6.1.2. Approach

In the algorithm we consider four binary matrices whose elements represent a transparent (0) or opaque (1) pixel, namely:

- S, Secret binary image
- R1, First Random grid
- R2, Second Random grid generated from S and R1
- Final, The Superimposed recovered image

In this algorithm, the images used were grayscale versions of images taken from a standard digital camera. The images were then converted into matrices for further computation.

6.1.3. Algorithms:

Algorithm for main():

Step 1: Start

Step 2: Read Image & store it in variable 'img'. Make 'img' into a NumPy array

Step 3: Calculate shape of 'img' and store it in 'l' and 'b'

Step 4: Create arrays 'secret', 'r1', 'r2' and 'final' of size l*b

Step 5: Binarize 'secret' with default thresholding. This means converting the image from grayscale to black and white. This is done by checking each element, if it is between 0 and 127, it is taken as 0 else 1.

Step 6: Create first random grid 'r1' using: create_r1(l,b)

Step 7: Create second random grid 'r2' using: create_r2(secret,r1,r2,l,b)

Step 8: Superimpose 'r1' and 'r2' using: superimpose(r1,r2,final,l,b)

Step 9: Stop

Algorithm for: create_r1(l,b):

Step 1: Make two for loops for variable 'i' from $1 \leq i \leq \text{width}$ and j from $1 \leq j \leq \text{height}$

In each iteration, perform $R1[i,j] = \text{random_pixel}(0,1)$

Algorithm for: create_r2(secret,r1,r2,l,b)

Step 1: Make two for loops for variable 'i' from $1 \leq i \leq \text{width}$ and j from $1 \leq j \leq \text{height}$

Check if $(B[i,j] == 0)$.

If yes, $R2[i,j] = R1[i,j]$

Else, $R2[i,j] = \text{complementOf}(R1[i,j])$

Algorithm for: superimpose(r1,r2,final,l,b)

Step 1: Make two for loops for variable 'i' from $1 \leq i \leq \text{width}$ and j from $1 \leq j \leq \text{height}$

Check if $((r1[i,j]==1 \text{ and } r2[i,j]==0) \text{ or } (r1[i,j]==0 \text{ and } r2[i,j]==1))$

If yes, $\text{final}[i,j] = 1$

Else, $\text{final}[i,j] = 0$

6.2. Implementation of multiple information hiding algorithm on square random grids

6.2.1. Introduction

This algorithm allows up to four secret images to be hidden on the four rotations of a square, namely 0, 90, 180 and 270. The advantage of this technique is that the individual shares reveal no information about the secret images. Only by combining the shares can the original images be revealed. The rotations provide an additional layer of security and randomness.

6.2.2. Approach

Four secret images are taken as input. These images must be square in shape and all of the same length and breadth.

In the encryption process, a random grid 'r1' is generated of the same dimensions. The random grid 'r2' is generated at 4 angles - 0, 90, 180 and 270 degrees by taking each of the 4 secret images and 'r1' respectively.

In the decryption process, the 4 random grids 'r2' generated are rotated by their respective angles and then superimposed with 'r1' using XOR operation.

In order to secure the process even further, a random number between 1 to 4 is chosen. According to this number, the random angle of the square to be used is chosen.

6.2.3. Algorithms:

Algorithm for: main():

Step 1: Start

Step 2: Read image 1 using `cv2.imread()`

Step 3: Resize image to standard 512*512 using `cv2.resize()`

Step 4: Initialize the length 'l' and breadth 'b' as $l = b = 512$

Step 5: Create numpy arrays:

`secret1 = np.zeros([l,b],dtype=int)`

final1 = np.zeros([l,b],dtype=int)

Step 6: Binarize the image as follows: secret1 = binarize_cv2(img1)

Step 7: Create first random grid r1 = create_r1(l,b)

Step 8: Create second random grid r2 = create_r2_at_0(secret1,r1,r2,l,b)

Step 9: Superimpose r1 and r2: final1 = superimpose_at_0(r1,r2,final1,l,b)

Step 10: Write final1 in file: write_in_file(final1, "image1_superimp.png")

Step 11: Calculate PSNR for final1: psnr1 = calc_psnr(secret1,final1)

Step 12: Repeat Step 2 to 11 for all 4 images

Step 13: Make a function to choose a random number from 1 to 4. According to the number, choose the random image to hide.

Step 14: Stop

Algorithm for: create_r2_at_0 (secret,r1,r2,l,b):

Step 1: For each pixel secret[i,j], $0 \leq i \leq l$ and $0 \leq j \leq r$,

Check if (secret[i,j] == 0)

If yes, $r2[i,j] = r1[i,j]$

else, $[i,j] = 1 - r1[i,j]$

Algorithm for: create_r2_at_90 (secret,r1,r2,l,b):

Step 1: For each pixel secret[i,j], $0 \leq i \leq l$ and $0 \leq j \leq r$,

Check if(secret[j,l-i-1] == 0)

If yes, $r2[j,l-i-1] = r1[i,j]$

else, $r2[j,l-i-1] = 1 - r1[i,j]$

Algorithm for: create_r2_at_180 (secret,r1,r2,l,b):

Step 1: For each pixel secret[i,j], $0 \leq i \leq l$ and $0 \leq j \leq r$,

Check if(secret[l-i-1,l-j-1] == 0)

If yes, $r2[l-i-1,l-j-1] = r1[l-j-1,i]$

else, $r2[l-i-1,l-j-1] = 1 - r1[l-j-1,i]$

Algorithm for: create_r2_at_270 (secret,r1,r2,l,b):

Step 1: For each pixel secret[i,j], $0 \leq i \leq l$ and $0 \leq j \leq r$,

 Check if(secret[l-j-1,i] == 0)

 If yes, $r2[l-j-1,i] = r1[j,l-i-1]$

 else, $r2[l-j-1,i] = 1 - r1[j,l-i-1]$

6.3. Implementation of multiple information hiding algorithm on same pair of rectangular random grids

6.3.1. Introduction

The previous algorithm used has the disadvantage of predictability of angles since a square can be rotated only in 4 ways to get the decrypted image. Hence, the following algorithm is used. In this, only a single pair of random grids is used. The advantage is that there is no limit on the number of secret images that can be hidden.

Here, R1 has breadth equal to the total breadth of all the secret images. R2 is generated using R1 for each of the secret images. Decryption can be performed by sliding and superimposing each R2 generated over R1.

6.3.2. Approach

In this approach, three images ($m=3$) are hidden on the same pair of rectangular random grid of width = $120*m$ and height = 200. Since m is 3, width = 360.

Each secret image is of the width = 120 and height = 200. The first random grid r1 is created with width $120*m$ and height 200. The second random grid r2 is created by using each image and the first random grid.

A random number k is selected from 1 to m . According to the value of k , the superimposed image is generated from r1 and r2.

Below is a visualization of the algorithm implemented.

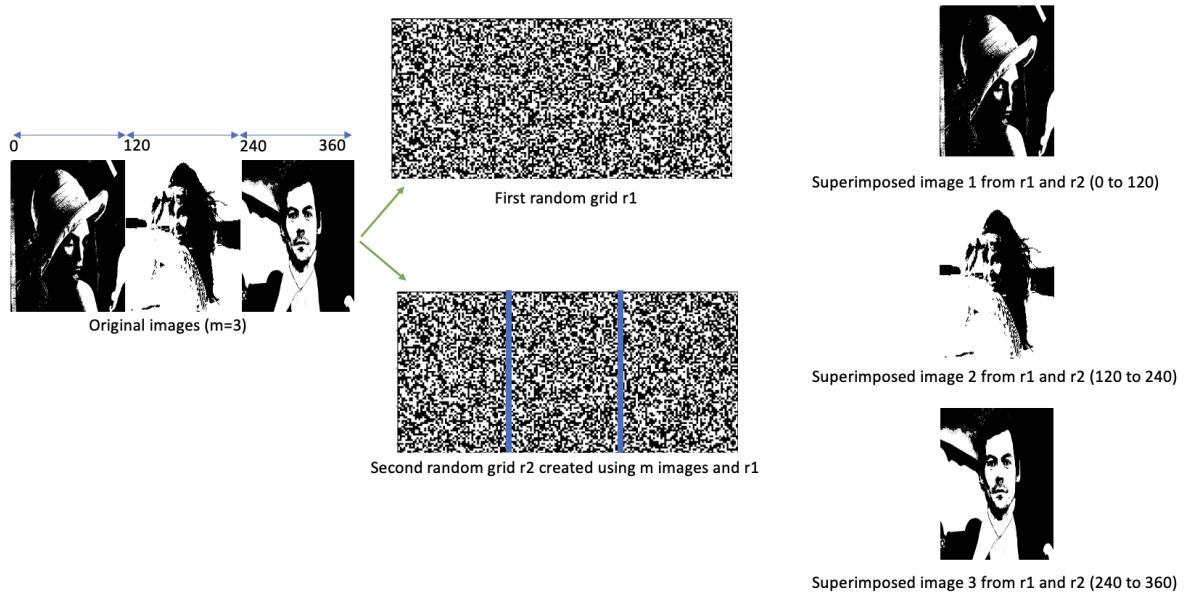


Figure 6.2. Flowchart for multiple information hiding algorithm on same pair of rectangular random grids

6.3.3. Algorithm

Algorithm for main():

Step 1: Start

Step 2: Read image 1 using `cv2.imread()`

Step 3: Resize image to standard 120*200 using `cv2.resize()`

Step 4: Initialize the length 'l' and breadth 'b' as $l = 200$ and $b = 120$.

Also initialize number of images 'm' = 3 and random number 'k' between 1 and m

Step 5: Create numpy arrays:

`secret1 = np.zeros([l,b],dtype=int)`

`final1 = np.zeros([l,b],dtype=int)`

Step 6: Binarize the image as follows: `secret1 = binarize_cv2(img1)`

Step 7: Repeat Step 2 to Step 6 for all m images

Step 8: Create first random grid `r1 = create_r1(l,b*m)`

Step 9: Create r2 at 0, r2 at 120 and r2 at 240 using all three secret images

Step 10: According to the value of k, display the k^{th} superimposed image

Algorithm for create_r2_at_0(secret,r1,r2,l,b,k):

Step 1: For each pixel secret[i,j], $0 \leq i \leq l$ and $0 \leq j \leq b$,

Check if (secret[i,j] == 0)

If yes, $r2[i,j] = r1[i,j]$

else, $[i,j] = 1 - r1[i,j]$

Algorithm for create_r2_at_120(secret,r1,r2,l,b,k):

Step 1: For each pixel secret[i,j], $0 \leq i \leq l$ and $b \leq j \leq (k-1)*b$,

Check if (secret[i,j-b]==0)

If yes, $r2[i,j] = r1[i,j]$

else, $[i,j] = 1 - r1[i,j]$

Algorithm for create_r2_at_240(secret,r1,r2,l,b,k):

Step 1: For each pixel secret[i,j], $0 \leq i \leq l$ and $(k-1)*b \leq j \leq k*b$,

Check if (secret[i,j-(k-1)*b]==0)

If yes, $r2[i,j] = r1[i,j]$

else, $[i,j] = 1 - r1[i,j]$

6.4. Implementation of multiple information hiding algorithm on circular cipher grids

6.3.1. Introduction

Earlier we used square shaped random grids for encryption which has a drawback of predictable angles. The rectangular random grid fixed this issue by making the angles unpredictable and setting no limit to the number of secrets. Later this rectangular grid was reshaped into a circular cypher grid.

6.4.2. Approach

This approach uses a circular cipher image with a radius of $r = 100$ px and $r' = 50$ px to hide text-based data. The same-dimension circular cipher random grid R1 will be created.

Each image is 300x300 pixels in size. The second random grid r2 is created by using the secret image and the first circular cypher random grid.

The superimposed (xor) image is generated from r1 and r2.

The implemented algorithm is displayed here.

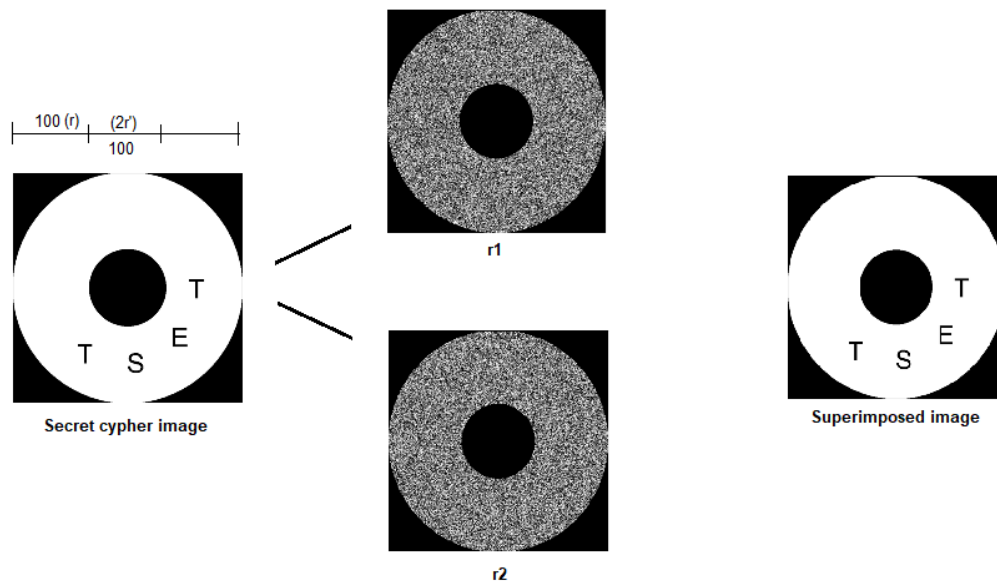


Figure 6.3. Flowchart for multiple information hiding algorithm on circular cypher grids

6.4.3. Algorithm

Algorithm for binarizing the image (img,l,b) :

Step 1: for i in range(l)
 for j in range(b)
 Check if (img[i,j] > 128 and img[i,j]<255)
 If yes, img[i,j]=1
 else, img[i,j]=0

Algorithm for create_r2 (secret,r1,r2,l,b,k):

Step 1: For each pixel secret[i,j], $0 \leq i \leq l$ and $(k-1)*b \leq j \leq k*b$,
 Check if (secret[i,j-(k-1)*b]==0)
 If yes, $r2[i,j] = r1[i,j]$
 else, $[i,j] = 1 - r1[i,j]$

Algorithm for XOR operation:

Step 1: for $0 \leq i \leq l$ and $0 \leq j \leq b$,
 Check if $((r1[i,j]==1 \text{ and } r2[i,j]==0) \text{ or } (r1[i,j]==0 \text{ and } r2[i,j]==1))$
 If yes, final[i,j] = 1
 else, final[i,j] = 0

Algorithm for main():

Step 1: Start

Step 2: Create blank image of 1:1 dimension using `image.new()`

Step 3: Create a circular cypher image of radius r and r' on the blank image using `draw()`

Step 4: Create numpy arrays:

`secret = np.zeros([l,b],dtype=int)`

`r2 = np.zeros([l,b],dtype=int)`

`final = np.zeros([l,b],dtype=int)`

Step 5: Binarize the image as follows: `secret = binarize_cv2(img)`

Step 6: Create first random grid `r1` using `create_r1(random_grid, mask)`

Step 7: Create `r2` using `create_r2(secret,r1,r2,l,b)`

Step 8: Superimposed image will be generated using `superimpose(r1,r2,final,l,b)`

6.5. Using Machine Learning to enhance Visual Cryptography using Random Grids

6.5.1. Background:

The usage of Machine Learning (ML) algorithms for Visual Cryptography (VC) using Random Grids is explored. This is due to the potential offered by Deep Learning and Neural Network algorithms for pattern recognition in images, natural languages, and other computer vision tasks. Specifically, Deep Learning algorithms are well-suited for pattern recognition in image applications, such as traffic management, number plate recognition, and more.

6.5.2. Introduction:

Deep Learning is a subset of ML that involves training Artificial Neural Networks with multiple layers to perform complex tasks such as Image Recognition, Speech Recognition, Natural Language Processing, and more.

Convolutional Neural Network (CNN) is a type of Deep Learning Algorithm that has application in image classification, object detection, and other computer vision tasks.

6.5.3. Basis for CNN in VC using RG:

One of the key usages of CNN is in imaging, where it can learn from a dataset of images and classify them into specific types. CNN works by learning features using raw image data and iterating/convolving over it multiple times through the Neural Network by applying different types of filters to recognize patterns. After training, the CNN model thus created can then be used to predict the correct label for a new image, as perceived by the human eye.

CNN can be used to improve the quality of the secret shares in VC using RG.

CNN models can be trained on a Dataset consisting of secret images combined with their corresponding shares to learn how to enhance the contrast and visibility of the shares. The trained CNN model can then be used to generate high-quality shares for new secret images.

6.5.4. Approach:

Typically a CNN model is created by using training and testing data from a given dataset split, say 80% of data as training data and the remaining 20% data as testing

Training data in the CNN needs two sets of data- X_{train} (input) and y_{train} (output/label). Then the model is tested using datasets X_{test} and y_{test} .

In this algorithm, the MNIST (Modified National Institute of Standards and Technology) dataset was used. The MNIST dataset consists of a large collection of handwritten digits (0-9)

Encoding:

The secret image is split as per the usual Random Grids algorithm for R1 and R2

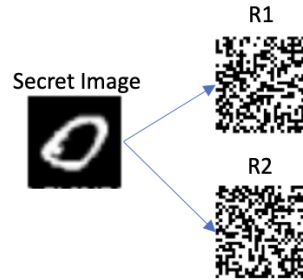


Figure 6.4. Encoding the Secret image to R1 and R2

CNN Model:

The CNN model is trained as follows:

- o **X_{train} :** This dataset consists of the secret image and R1 share
- o **y_{train} :** This dataset consists of R2 share

The model is then tested using X_{test} , y_{test} and y_{pred} (predicted output).

- o **X_{test} :** This consists of secret image and R1 share
- o **y_{test} :** This consists of R2 share
- o **y_{pred} :** This is the predicted R2 share from the CNN model

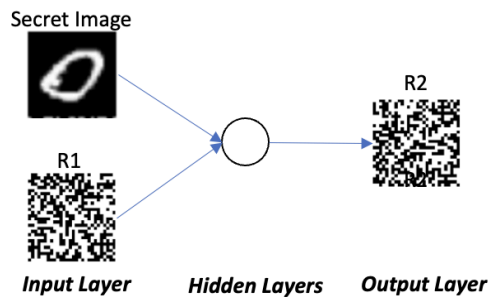


Figure 6.5. High Level Architecture of the Training Model

Decoding:

The R2 share generated without CNN (y_{test}) and R2 share generated with CNN (y_{pred}) is decoded.

- o $decoded_pred = r1 \text{ (XOR) } (y_{pred})$
- o $decoded_test = r1 \text{ (XOR) } (y_{test})$

Both of these decoded images are then evaluated by finding their PSNR values.

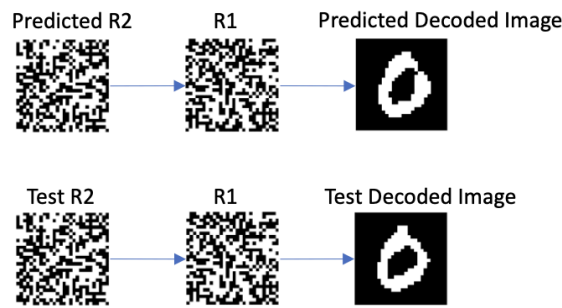


Figure 6.6 Decoding of output (predicted and test)

6.5.5. Algorithm:

Step 1: Start

Step 2: Load training and test data of handwritten digits from MNIST dataset. Store them into X_train, X_test, y_train, y_test

Step 3: Normalise X_train and X_test by dividing it by 255.0. Binarize the floating point values. If the pixel value is less than 0.5, assign it as 0. Else assign it as 1.

Reshape both X_train and X_test to (-1,784)

Step 4: Make first secret share (random grid) r1

Step 5: Make the second secret share (random grid) r2_train using the X_train and r1 using the above given algorithm.

Repeat this for r2_test

Step 6: Concatenate r1 to each row of X_train and X_test. Then reshape both X_train and X_test to (-1, 56, 28, 1)

Step 7: Assign y_train = r2_train and y_test = r2_test

Reshape both to (-1,28,28,1)

Step 8: Define a Sequential() model and add the layers:

- o Conv2D(32, (3, 3), activation='relu', input_shape=(56,28,1))
- o Conv2D(32, (3, 3), activation='relu')
- o MaxPooling2D(pool_size=(2, 2))
- o Conv2D(64, (3, 3), activation='relu')
- o Conv2D(64, (3, 3), activation='relu')
- o MaxPooling2D(pool_size=(2, 2))
- o Flatten()
- o Dense(128, activation='relu')
- o Dense(28*28, activation='sigmoid')
- o Reshape((28, 28, 1))

Step 9: Compile the model with binary cross entropy loss and Adam optimizer

Step 10: Add data augmentation to the data

Step 11: Fit the model to the training data

Step 12: Generate predicted images from model into y_pred

Step 13: Generate decoded images:

```

decoded_pred = r1 (XOR) (y_pred)
decoded_test = r1 (XOR) (y_test)

```

Step 14: Generate the PSNR ratio for decoded_pred and decoded_test

Step 15: Plot X_test, r1, decoded_pred and decoded_test

Step 16: Stop

6.5.6. Implementation

The following parameters were used in the Sequential model

- **Input shape:** (56, 28, 1)
- **Output shape:** (28, 28, 1)
- **Activation Functions used:**
 - **Rectified Linear Unit** - It introduces non-linearity by outputting the input if it is positive, and zero otherwise.
 - **Sigmoid** - It is used in the dense layer of CNNs that maps the output values to a range between 0 and 1, enabling the interpretation of the output as probabilities or continuous values.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 54, 26, 32)	320
conv2d_1 (Conv2D)	(None, 52, 24, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 26, 12, 32)	0
conv2d_2 (Conv2D)	(None, 24, 10, 64)	18496
conv2d_3 (Conv2D)	(None, 22, 8, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 11, 4, 64)	0
flatten (Flatten)	(None, 2816)	0
dense (Dense)	(None, 128)	360576
dense_1 (Dense)	(None, 784)	101136
reshape (Reshape)	(None, 28, 28, 1)	0

Figure 6.7. Summary of the Sequential Model

During Compilation of the model, the following parameters were used:

- **Loss Function**
 - **Binary Cross Entropy** - It is a loss function used in binary classification tasks where the model aims to minimize the difference between predicted and actual outputs by measuring the dissimilarity of their logarithmic probabilities.
- **Optimizer:**

- **Adam** - It is an optimization algorithm commonly used in deep learning that adapts the learning rate dynamically based on the gradients, resulting in faster convergence and improved performance.
- **Performance Metrics used:**
 - **Accuracy** - It is used to evaluate the performance of the model by calculating the proportion of correctly predicted samples.

During Training and fitting of the model, the following parameters were used:

- **Epochs: 5**
It specifies the number of times the model will be trained on the entire training dataset.
- **Batch Size: 128**
It determines the number of training examples used in one iteration during training.
- **Steps Per Epoch: $X_train.shape[0] \% 128$**
It defines the number of batches that will be processed per epoch in the training set.
- **Validation Steps: $X_train.shape[0] \% 128$**
It defines the number of batches that will be processed per epoch in the validation set.

7. RESULTS AND ANALYSIS

7.1. Performance Metrics

7.1.1. Pixel wise accuracy

- This metric measures the percentage of correctly reconstructed pixels in the decrypted image compared to the original image.
- It is the basic measurement of reconstruction quality.

Algorithm:

Step 1: Load the original and reconstructed images: `Image.open('image_file')`

Step 2: Conversion of images to numpy array: `np.array('image_name')`

Step 3: Total number of matching pixels: `np.sum(original_array == reconstructed_array)`

Step 4: The total number of pixels: `original_array.shape[0] * original_array.shape[1]`

Step 5: Calculation of pixel-wise accuracy percentage:

$$\text{Accuracy} = (\text{matching_pixel} / \text{total_pixel}) * 100$$

7.1.2. Peak Signal-to-Noise ratio

- It is the ratio between the original image and the noise (encrypted shares) in the decrypted image.
- Formula: $10 * \log_{10}(\text{max_pixel_value}^2 / \text{MSE})$,
MSE = Mean square error between the original and reconstructed images.
- Higher PSNR indicates better image quality.
- Assuming each pixel is represented using a single bit (binary image), the maximum possible PSNR value is 6.02 dB.

Algorithm:

Step 1: Load the original and reconstructed images: `Image.open('image_file')`

Step 2: Convert the images to numpy array: `np.array('image_name')`

Step 3: Calculate the mean squared error (MSE):

$$\text{mse} = \text{np.mean}((\text{original_array} - \text{reconstructed_array}) ** 2)$$

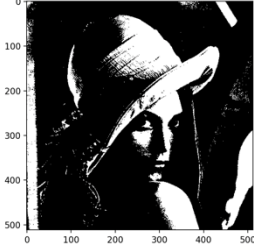
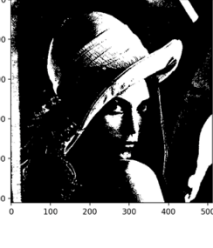
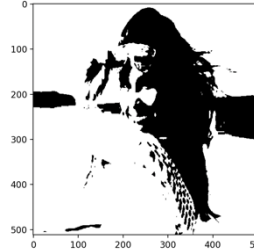
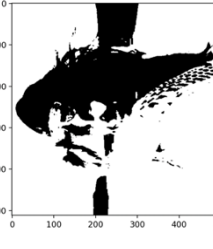
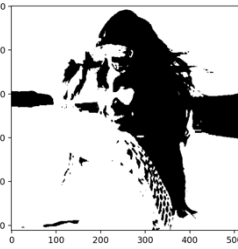
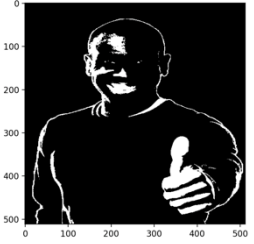
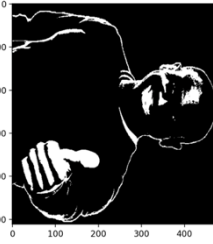
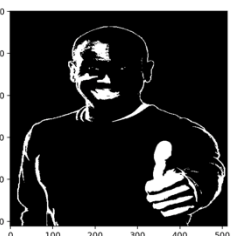
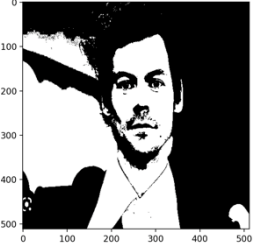
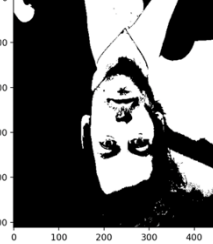
Step 4: Calculate the maximum pixel value: `max_pixel_value = np.max(original_array)`

Step 5: Calculate the PSNR: $10 * \log_{10}((\text{max_pixel_value} ** 2) / \text{mse})$

7.2. Results and Analysis

7.2.1. Results of Algorithm on Square Random Grids

Table 7.1: Representation of input images, output images and PSNR for the 4 angles

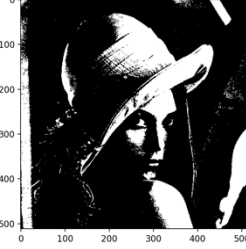
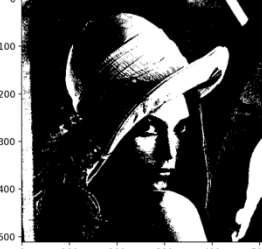
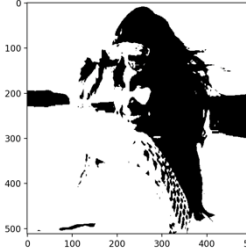
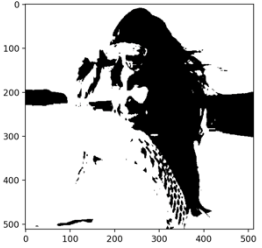
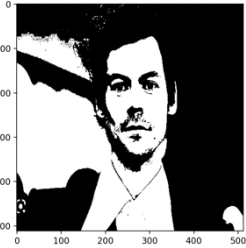
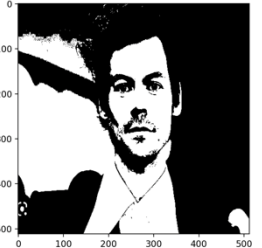
Angle	Binarized Input Image	Superimposed image	Superimposed rotated Output	PSNR	Pixel-wise accuracy
0°				6.717	0.785
90°				1.447	0.278
180°				10.905	0.918
270°				4.090	0.607

On using square random grids, up to four images can be encoded at a time on each of the four rotations of a square. This has the advantage of hiding multiple secrets. However, the disadvantage of this algorithm is the easy prediction of the angles of a square.

Here, images captured by a standard digital camera were used. The PSNR ratios were in the range of 1 to 11.

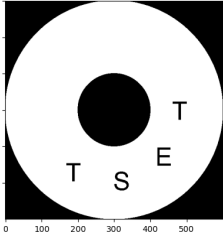
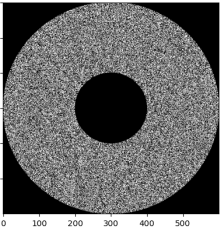
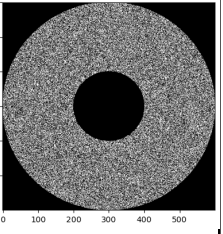
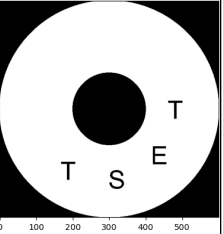
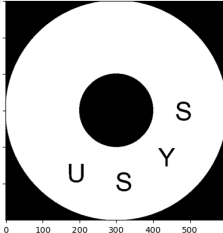
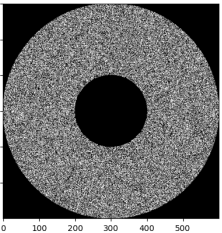
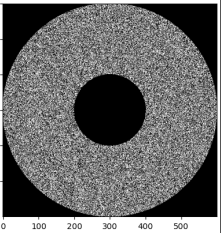
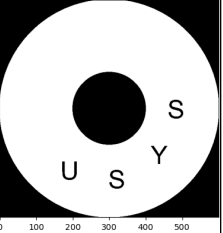
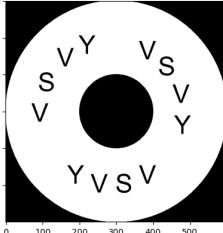
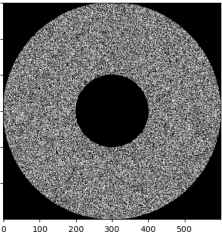
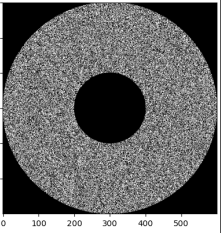
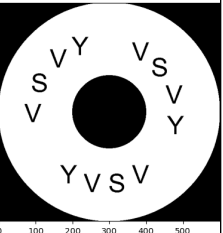
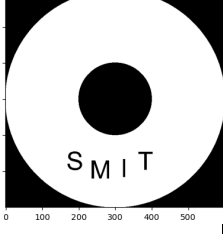
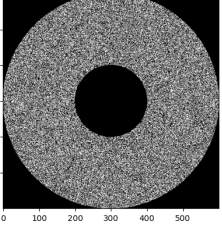
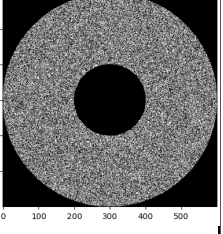
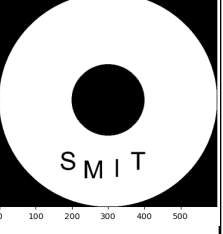
7.2.2. Results of Algorithm on same pair of Rectangular grids

Table 7.2: Representation of input images, output images and PSNR for various angles

Angle	Binarized Input Image	Superimposed Output Image	PSNR	Pixel-wise accuracy
0°			6.717	0.785
120°			1.447	0.278
240°			4.090	0.607

7.2.3. Results of Circular grids

Table 7.3: Representation of input images, random grids, output images and PSNR for the 4 angles

Angle.	Binarized Input Image	Cypher grid r1	Generated r2 using secret and r1	Superimposed image	PSNR
0°					4.569
90°					4.633
180°					4.807
270°					4.911

7.2.4. Results of using CNN to generate better quality random grids

Table 7.4: Comparison of Prediction vs Test Decoded images and their PSNR

Sl No.	Secret Image	Predicted Decoded Image	PSNR of Predicted Decoded Image	Test Decoded Image	PSNR of Test Decoded Image
1			9.353		9.305
2			10.682		10.618
3			8.080		8.044

From the table, it is seen that the PSNR of Predicted Image is higher than the Test Image.

Measuring Loss and Accuracy over the Epochs

Training loss - It refers to the loss or error of the model on the training dataset during each epoch. It represents the discrepancy between the predicted outputs of the model and the actual ground truth labels in the training data. The training loss is used to update the model's weights and improve its performance through gradient descent.

Training accuracy - It refers to the accuracy of the model on the training dataset during each epoch. It indicates how well the model is performing on the data it has been trained on.

Validation loss - It is the loss or error of the model on the test dataset during each epoch. It is calculated using the same loss function as the training loss and serves as a measure of how well the model generalizes to new, unseen data.

Validation accuracy - It represents the accuracy of the model on a separate validation dataset during each epoch. The validation dataset is used to evaluate the model's performance on unseen data and serves as a measure of how well the model generalizes to new examples.

Table 7.5: Comparing the loss and accuracy for the initial and final Epoch

Parameter	1st Epoch	5th Epoch
Training loss	0.2232	0.1258
Training accuracy	0.8903	0.9429
Validation loss	0.1689	0.1209
Validation accuracy	0.9208	0.9455

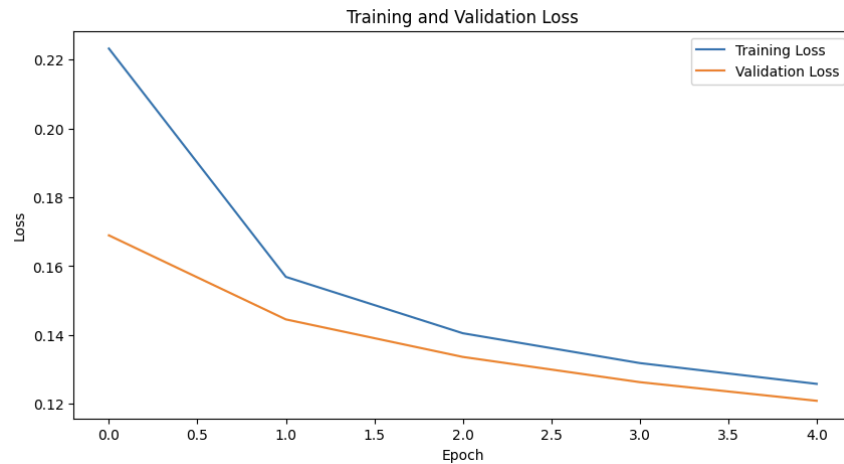


Figure 7.1: Visualization of the training and validation loss over the epochs.

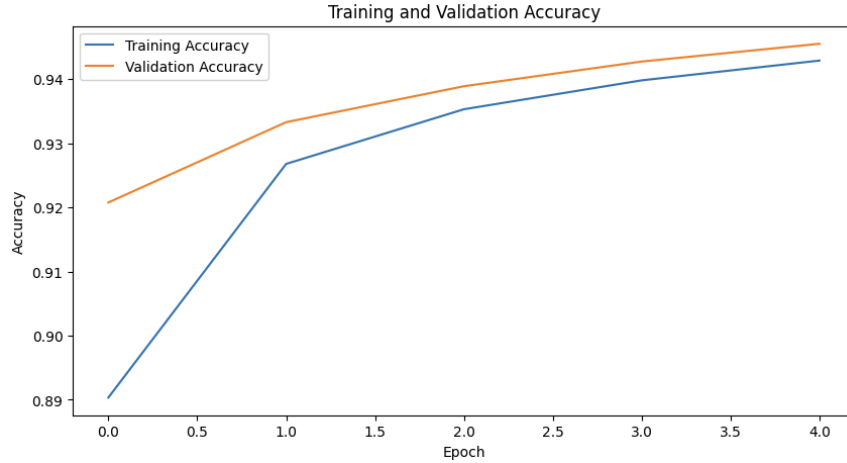


Figure 7.2: Visualization of the training and validation accuracy over the epochs.

8. CONCLUSION

The following proposed schemes were achieved.

- Information hiding on square random grids
- Information hiding on same pair of rectangular random grids
- Information hiding on circular random grids
- Using Machine Learning Approach to generate better quality random grids

The scheme for Multiple Information Hiding was achieved. This method can be used to transmit many secret images at once. Since the grids are unrelated, the information hidden in one grid provides no clues or information about the other grids. Hence, it decreases the probability of attacks due to lower predictability.

The scheme to use Machine Learning Algorithms, specifically Convolution Neural Networks, to generate better quality random grids was achieved. There was a marginal improvement in PSNR of predicted images using CNN when compared to test images.

9. LIMITATIONS AND FUTURE SCOPE

9.1 Limitations

The primary limitation of the scheme is that it only works on binary images. The image datasets used for all the algorithms are grayscale which are then converted to binary values for the scheme to work.

Secondly, the circular grid scheme has the problem of different sizes of the sector pixels. Also the PSNR of the decrypted image decreases with increase in the number of secrets.

Thirdly, in the CNN approach for generation of better quality random grids, the PSNR of the prediction was only slightly higher than the test dataset.

9.2 Future Scope

Firstly, the schemes can be modified to use color images as well.

Secondly, in the CNN approach, the number of layers, types of layers and Activation Functions in the model can be changed in order to achieve even higher quality random grids.

Thirdly, a combination of CNN approach and multiple information hiding can be used to simultaneously increase the quality of decoded images while also reducing the probability of attacks.

10. GANTT CHART

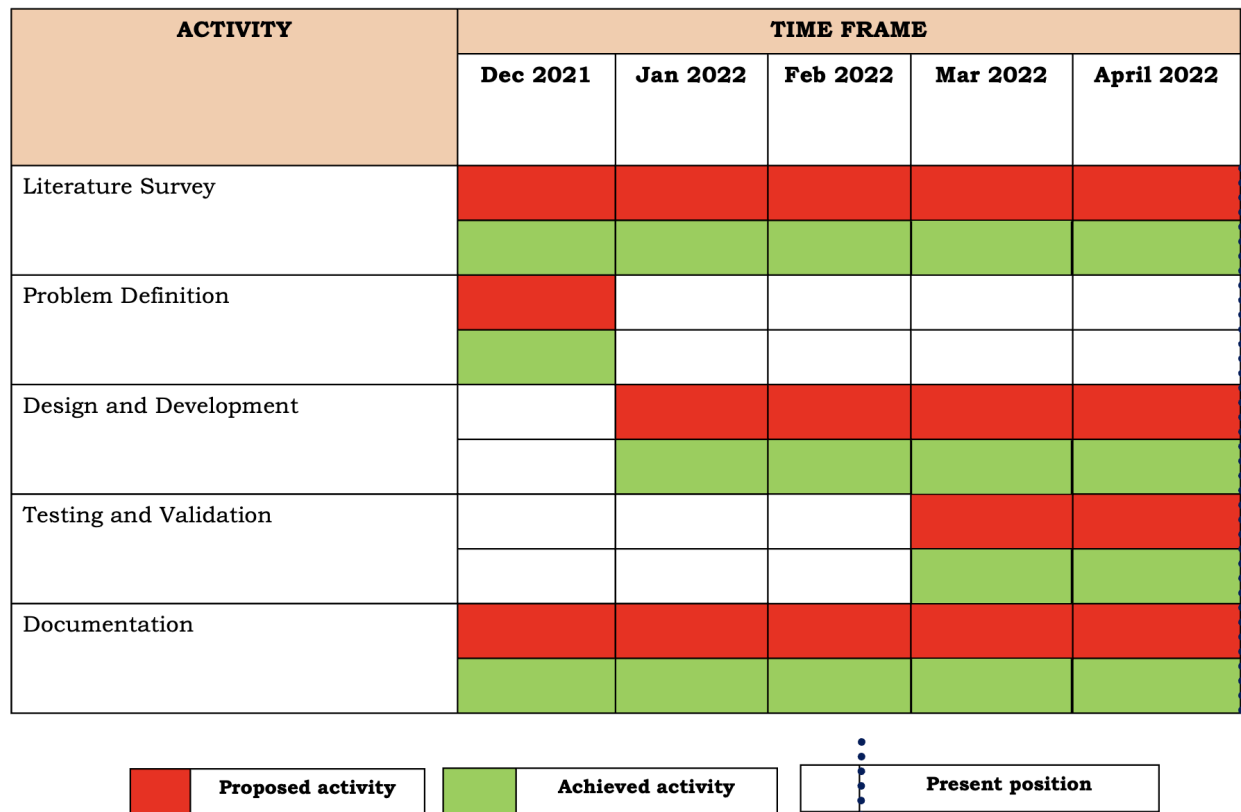


Figure 10.1: Gantt Chart

11. REFERENCES

- [1] Naor, Moni, and Adi Shamir. "Visual cryptography." Advances in Cryptology—EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings 13. Springer Berlin Heidelberg, 1995.
- [2] Kafri, Oded, and Eliezer Keren. "Encryption of pictures and shapes by random grids." Optics letters 12.6 (1987): 377-379.
- [3] Saha, Pratarshi, Sandeep Gurung, and Kunal Krishanu Ghose. "Hybridization of DCT based steganography and random grids." International Journal of Network Security & Its Applications 5.4 (2013): 163.
- [4] Chen, Tzung-Her, and Kuang-Che Li. "Multi-image encryption by circular random grids." Information Sciences 189 (2012): 255-265. Tzung-Her Chen and Kuang-Che Li "Multi-image encryption by circular random grids" Elsevier, 2011
- [5] Deepa, G. "The comparative study on visual cryptography and random grid cryptography." IOSR Journal of Computer Engineering 12.2 (2013): 04-14.
- [6] Sonar, M., Mr. (2022, September 29). Visual cryptography introduction. <https://www.geeksforgeeks.org/visual-cryptography-introduction/>

- [7] Mandeep Kaur, Surender Singh, Manjit Kaur, "Computational Image Encryption Techniques: A Comprehensive Review", *Mathematical Problems in Engineering*, vol. 2021, Article ID 5012496, 17 pages, 2021. <https://doi.org/10.1155/2021/5012496>
- [8] Wu, L. and Chen H. A Study On Visual Cryptography, Master Thesis, Institute of Computer and Information Science, National Chiao Tung University, Taiwan, 1998.
- [9] Gurung, Sandeep, et al. "Multiple information hiding using cubical approach on random grids." *International Journal of Computer Network and Information Security* 7.11 (2015): 54-63.
- [10] Sharma, Kartik, et al. "Hiding data in images using cryptography and deep neural network." *arXiv preprint arXiv:1912.10413* (2019).
- [11] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9, 611–629 (2018).
<https://doi.org/10.1007/s13244-018-0639-9>