IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 24, Issue 3, Ser. II (May. –June. 2022), PP 49-59
www.iosrjournals.org

Digital Image Encryption based on Gauss Map and Gabor Transformation

Srushti Gandhi[1], Ravi Gor[2]
[1]*Research Scholar, Department of Mathematics, Gujarat University, Gujarat, India*
[2]*Department of Mathematics, Gujarat University, Gujarat, India*

Abstract
In the recent decades, image encryption has fascinated many researchers and scientists. However, numerous studies have been made with different methods; and novel algorithms have been proposed to improve the secure digital image encryption systems. Nowadays, chaotic methods have been initiated in several fields, such as the design of cryptosystems and image encryption. The chaotic methods-based digital image encryption system is a quite novel method. In this technique, the chaos sequences are used for encryption of images. It is a kind of highly-secured image encryption method. A novel technique is proposed for digital image encryption and improved earlier algorithms. The simulation and theoretical analysis demonstrate effectiveness and usefulness of the proposed scheme and clearly reveal that this technique is a suitable choice for the actual or real image encryption.

***Key Words*:** *Digital Image, Encryption, Chaotic Method, Chaos Sequence, Gauss Map, Coupled Map Lattice, Gabor transformation.*

Date of Submission: 02-06-2022 Date of Acceptance: 15-06-2022

I. Introduction

In modern days, an effective technique for image encryption has been a placing place for studies. It is extensively recognised and accepted as a valuable technique for easy transmission. Every image encryption, established by the rules, is supposed to generate top-first-rate of noisy image to keep the facts secret. In addition, image encryption is the most beneficial component for guaranteeing classified or catagorised transmissions and image capabilities over the Internet. The digital verbal exchange has become larger by using the immediate improvement of Internet technology. People can send digital images over the internet anytime and anywhere. This has led to an increase in the growth of digital image security by encryption technique.

The various strategies representing digital image encryption in studies are correlated to the ever-increasing need and necessity of security. Image encryption based on the chaos method is a novel encryption technique for images in which a random chaos series is applied to encrypt the image as a powerful way for solving the stubborn issues of distinctly comfortable and explicitly rapid image encryption. In the last few years, different versions of the chaos method have been provided.

Presently, four methods are implemented for image encryption, applying numerous concepts personally and accomplishing the equal targets. These four principles are (i) Sharing and Secret Segmentation, (ii) Sequential Permutation, (iii) Chaotic Dynamical Systems, and (iv) Cutting-Edge Cryptography, every with particular features. First of all, the plain-text images are divided into a few blocks employing the proposed method. Then the correlation coefficients are determined. In the first step, the original image is processed using the Gauss map to obtain a diffused image; which is then transformed into a confused image when it is processed under the coupled lattice map (CLM) to generate two random serieses, then they are XORed with the pixel values of the image.

In the end, the complete image is permuted by two random sequences produced from the chaotic maps.

II. Literature Review

Patidar et al. [4] (2011) represented some other dynamic pseudorandom permutation-substitution outline which is solely based on the chaos theory for the image encryption. It was a lossless symmetric block cipher and was designed particularly for colour images. It may also be applied for grayscale images.

Xu et al. [7] (2014) exploited the synchronizing fractional chaotic systems for image encryption.

Enayatifar et al. [1] (2014) have generated the Deoxyribo-Nucleic Acid (DNA) sequence and used hybrid genetic algorithms were used for image encryption. They presented a new image encryption algorithm using a hybrid model of a genetic algorithm (GA), DNA masking and a logistic map.

Li et al. [3] (2018) represented multiple-image encryption via the sturdy chaotic map in the wavelet rework domain. In this image, first, the Discrete Wavelet Transform (DWT) is used to decompose the authentic images getting used and reassemble the lower frequency components as the direct images (predicted images). Then the direct image is transformed into absolutely scrambled one via Arnold's cat map. Similarly decomposition of the scrambled image and the resulting block image are hired one by one to combine with the amplitude parameter of the Robust Chaotic Map (RCM) for generating a keystream in every diffusion technique.

Satish et al. [6] (2018) offered a skeleton to encrypt an image through the Logistic Map. It could scramble the image pixels. Thus, the resulting cipher image can be XOR-encrypted when the output is divided into numerous frequency coefficients by integer wavelet decomposition. The Logistic Map is used to shuffle the image pixels ensuing low-frequency coefficient wavelet, and all of the frequency coefficient wavelets can be integrated through Inverse Integer Wavelet Transformation. Their approach was based on the chaos principle for the digital photograph encryption. Since the chaos-based image encryption approach has some difficulties and complexities, which include restrained accuracy, the encryption process of image is split into spatial and revised domain encryption.

Hafsa et al. [2] (2021) proposed an image encryption model which is a complex chaos-based pseudorandom number generator and modified advanced encryption standard. the overall system was created on the Altera Cyclone-III board. Their findings exposed that the cryptographic algorithm was faster and could tolerate some kind of attacks.

Pourasad et. al. [5](2021) proposed a fast, highly-secured and improved method for digital image encryption technique that used random chaos sequences for encrypting images. Limited accuracy is one of the limitations of this technique. They generated the chaos sequence and wavelet transform value to find the gaps.

In the last few years, some image encryption schemes by the frequency domain and the spatial domain have been introduced. Spatial domain methods operate directly on the pixels of a plain image. This method is widely used because of its high-speed encryption. The transform domain encryption is used, considering some of the characteristic properties of digital images as a strong correlation between high redundancy and nereby pixels.

Based on the results in the planned skeleton, the properties are improved with wider chaotic ranges and more dynamic chaotic behavior. The integration of chaos sequence and gabber transformation value and image encryption algorithm all together are useful. Such algorithms are replicated by analyzing algorithms to control gaps. Therefore, the algorithm will be boosted. This method uses two chaotic systems that can also use a fundamental nonlinear equations to denote the chaotic behavior.

III. Terminologies Used In The Current Work

Chaos and Transformation theory

Chaos theory has been used for several years in cryptography. In the past few years, chaos and nonlinear dynamics have been used to propose hundreds of cryptographic primitives. These algorithms contain image encryption algorithms, hash functions, secure pseudo-random number generators, stream ciphers, watermarking and steganography. In a broader perspective, the similarities amongst the chaotic maps and the cryptographic systems, without losing their generality, are the main motivations for the design of chaos-based cryptographic algorithms. One type of encryption, secret key or symmetric key, depends on diffusion and confusion, which may well be proved by the chaos theory.

At the present time, theories of chaos and transformation have evolved into novel currencies in the social sciences. Image transformation is a technique that simplifies image processing and refines the performance of image processing. Image correction and improvement indicate highlighting and sharpening certain definite features. This includes the contours of the outline, edges, and contrast of the image for displaying, observing, or further analyzing and processing the image.

Chaos sequence based on Gaussian map

The Gauss map[8] (also known as Gaussian map or mouse map), is a nonlinear iterated map of the reals into a real interval given by the Gaussian function:

$$G(x) = e^{-\alpha x^2} + \beta$$

Where, α and β are real parameters,

$x = \frac{1}{\sqrt{2\alpha}}$,

$\beta = 0 \Longrightarrow x_{n+1}$ reduces to $\beta + 1$.

$\Longrightarrow G(x)$ is the constant map.

$\beta \neq 0 \Longrightarrow G(x)$ is not bounded.

which is stated in state equation form as

$$x_{n+1} = e^{-\alpha x_n^2} + \beta, n = 1, 2, \ldots$$

where $x_n = \frac{1}{\sqrt{2\alpha}}$ and $\alpha, \beta > 0$ are well-known bifurcation parameters. Here, x_n represents the system's state at time n while x_{n+1} indicates the following state; and n shows the discrete-time. By repeated iteration of G, a sequence of points $\{x_n\} \to \infty$ is increased, known as an orbit.

The Gauss map is also one of the well-recognised and commonly active maps in generating chaotic sequences:

$$x_{n+1} = \begin{cases} 0; & x_n = 0 \\ \frac{1}{x_n} \, mod 1; & otherwise \end{cases}$$

Now, the diffusion algorithm key is selected, for which the actual y, the primary iteration of the Gaussians, is with parameter m. For different primary situations, two gaussian maps are utilized for executing the repetition operation. Moreover, the values of the state of two gaussian maps are measured dynamically.

Utilizing this technique, the chaotic sequences are generated. The entire procedure is as follows:

Place an image with the size of $m \times n$, the data matrix of R. Turn R into a one-dimensional matrix with the length of $m \times n$. Put $R_1 = \{r_1, r_2, r_3, \ldots, r_{m \times n}\}$, and put $P_1 = \{p_1, p_2, p_3, \ldots, p_{m \times n}\}$, as the encrypted 1D matrix.

The step-by-step procedure of the encrypted algorithm is as follows:

Step - 1: Two chaotic sequences, $x = \{x_1, x_2, x_3, \ldots, x_{m \times n}\}$, are formed by two one-dimensional gaussian maps. Place the two Gaussian maps system parameters as primary values as $x_1(0)$ and $x_2(0)$, respectively.

Step - 2: For every iteration, compare $x_1(i)$ and $x_2(i)$, $i = 1, 2, \ldots, m \times n$ and choose the one that is numerically larger.

Step – 3: Perform the Exclusive-OR (XOR) operation for the sequences produced by Step 2 using the pixels of the original image.

Thus, a diffused image is obtained.

Kinetics of Coupled Map Lattice

One of the extremely well-known teachings of ways within the theory of space-time chaos is formed by using the Coupled Map Lattice (CML). The CMLs are used in cryptography, physics, economics, steganography, and biology. They have an extensive position in image encryption algorithms. Then, we used a two-dimensional hyper-chaotic map CML to try pixel area. It can successfully and effectively enlarge the key area. It increases the functionality of anti-decryption.

CML statement is as:

CML is used in cryptography, chemistry, biology, biochemistry, genetics, physics, economics, steganography etc. One of the most well-known doctrines within the theory of space-time chaos is formed using a CML. They have a wide place in image encryption algorithms. Then, we used the two-dimensional hyper-chaotic map CML to try out the pixel area. It can expand the key area successfully and efficiently. It increases the efficiency of anti-decryption. The CML statement is as follows:

$$x_{n+1} = 1 - \alpha(x_n^2 + y_n^2)$$
$$y_{n+1} = -2\alpha(1 - 2\beta)x_n y_n$$

The digital images possess the digital matrix features for scrambling the location of pixels; hence, considering a random image, the impact of confidentiality is achieved.

The procedure of the algorithm of encryption is as follows:

Step - 1: The chaotic sequences $x_1, x_2 = \{x_1, x_2, x_3, \ldots, x_m\}$ are produced with the length of m, and $y_1, y_2 = \{y_1, y_2, y_3, \ldots, y_n\}$ with the length of n similar to CML chaos mapping.

Step - 2: x, y chaotic sequences are arranged in rising sequences, producing position sequences w_2, w_3.

Step – 3: In the last step, the pixel confusion is performed, using w_1, w_2 as the row, and column sequences of the data matrix R.

$$R(i, 1) = R(w_1(i, w_2(j)))$$

Gabor Transformations and Inverse Gabor Transformations

Gabor Transformation

The Gabor transform is a special case of the short-time Fourier transform. It is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. The function to be transformed is first multiplied by a Gaussian function, which is also known as a window function, and then the resulting function is with a Fourier transform to derive the time-frequency analysis. The window function

means that the signal near the time being analysed will have a higher weight. The Gabor transform of a signal $x(t)$ is defined by the below given formula:

$$G(\tau,\omega) = \int_{-\infty}^{\infty} x(t) e^{-\pi(t-\tau)^2} e^{-j\omega t} dt$$

The Gaussian function has an infinite range and it is impractical for implementation. However, a level of significance can be chosen (for instance 0.00001) for the distribution of the Gaussian function.

$$\begin{cases} e^{-\pi a^2} \geq 0.00001; & |a| \leq 1.9143 \\ e^{-\pi a^2} < 0.00001; & |a| > 1.9143 \end{cases}$$

Outside these limits of integration ($|a| > 1.9143$) the Gaussian function is small enough that it can be ignored. Thus, the Gabor transform can be satisfactorily approximated as below:

$$G(\tau,\omega) = \int_{-1.9143+\tau}^{1.9143+\tau} x(t) e^{-\pi(t-\tau)^2} e^{-j\omega t} dt$$

This optimised simplification makes the Gabor transform quite practical and sufficiently realizable.
The window function width can also be varied to optimize the time-frequency resolution trade-off for a particular application by replacing the $-\pi(t-\tau)^2$ with $-\pi a(t-\tau)^2$ for some chosen α.

Inverse Gabor Transformation

The Gabor transform is invertible. The original signal can be recovered by the following equation:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G_x(\tau,\omega) e^{j\omega t + \pi(t-\tau)^2} d\omega$$

IV. Proposed Algorithm

The four steps for executing the proposed algorithm are given below.

Step - 1: An image G is arranged. The image's size is set to $m \times n$. Additionally, a data matrix R is placed. By evaluating two gaussian maps, a chaotic sequence is generated. Making XOR with the primary image, the diffusion is completed.

Step - 2: For the diffused image in step1, the Gabor decomposition is performed and then the Gabor coefficient is extracted, listed as ca1.

Step – 3: Utilizing the CML, the chaotic sequence is produced, and with ca1 established in step 2, the position confusion is performed.

Step – 4: In the last step, the confused image can be reconstructed by Gabor. Thus, the encrypted image is obtained.

The inverse operations of the encryption are known as the decryption algorithm.
System parameters and the primary value of the chaotic sequences in the image encryption and image decryption are reliable.

V. Example

For digital images:
Using python,
image array: <PIL.JpegImagePlugin. JpegImageFile image mode=RGB size=450x337 at 0x12C49E757C0>
Image shape: (337, 450, 3)
Image mode: RGB
Image size: (450, 337)

Figure 1: Digital Image

The Image array/in matrix form:

```
[[[98  95  50                                  [125 123 137
 80  84  33          ... ...                   132 132 132
 82  97  38]                                    158 158 158]]

[[86  83  38                                   [111 111 123
 74  78  27          ……                        143 144 146
 76  91  32]                                    167 168 170]]

[[88  84  39                                   [76  79  88
 86  88  39          ……                        155 160 163
 86 100  41]                                    157 162 165]]

   ……                ……                        ……
   ……                ……                        ……

[[21  39 113                                   [[89  99 101
  0  17  89          ……                         61  95  60
  0   8  77]                                    61  95  60]]

[[15  31 116                                   [94 104 105
  1  18 100          ……                         72 107  67
  0  16  94]                                    72 107  67]]

[[24  52 126                                   [105 120  97
 18  42 114          ……                         86 124  73
  8  28  97]                                    81 119 68]]]
```

Total pixel value: 454950

Gauss Map

Step - 1: Two chaotic sequences, $x = \{x_1, x_2, x_3, \ldots., x_{m\times n}\}$, are formed by two one-dimensional gaussian maps. Place the two gaussian maps system parameter as a primary value as $x_1(0)$ and $x_2(0)$, respectively.

a =	0.1738604485237788	b =	44.3344143735636
a =	0.2754932126159704	b =	70.25076921707245
a =	0.08329237286291546	b =	21.239555080043445
a =	0.3944273249663889	b =	100.57896786642917
a =	−0.1530865670751299	b =	−39.03707460415813
a =	0.30857079505619633	b =	78.68555273933006
a =	0.01582147659128863	b =	4.0344765307786
a =	0.43158169732419616	b =	110.05333281767003
a =	−0.21919380230387192	b =	−55.89441958748734
a =	0.19453405764281118	b =	49.60618469891685
a =	0.23988619526314492	b =	61.17097979210195
a =	0.15460141994454668	b =	39.4233620858594
a =	0.30625989283292177	b =	78.09627267239505
a =	0.020535657315830735	b =	5.236592615536837
a =	0.43061174247055645	b =	109.80599432999189
a =	−0.21754067356297485	b =	−55.47287175855859

a =	0.19765787593441775	b =	50.402758363276526
a =	0.23430056454643178	b =	59.746643959340105
a =	0.1655200266850929	b =	42.20760680469869
a =	0.28912378448163356	b =	73.72656504281656
a =	0.05552603339093387	b =	14.159138514688138
a =	0.41567050729220956	b =	105.99597935951344
a =	−0.19156644474421664	b =	−48.84944340977524
a =	0.24514533194839683	b =	62.51205964684119
a =	0.14423979223541916	b =	36.78114702003189
a =	0.32172714227419497	b =	82.04042127991971
a =	−0.01094739039182957	b =	−2.79158454991654
a =	0.4323207076528459	b =	110.2417804514757
a =	−0.22045055478001485	b =	−56.21489146890379
a =	0.19215204787135698	b =	48.998772207196026
	… … … … … … … … … … … …		… … … … … … … … … … … … …

… … … … … … … … … … … …

Step - 2: For every iteration, compare $x_1(i)$ and $x_2(i)$, $i = 1, 2, \dots, m \times n$ and choose one that is numerically larger.

44.3344143735636,	70.25076921707245,	21.239555080043445,
100.57896786642917,	−0.1530865670751299	78.68555273933006
4.0344765307786,	110.05333281767003,	−0.21919380230387192,
49.60618469891685,	61.17097979210195,	39.4233620858594,
78.09627267239505,	5.236592615536837,	… … … … … … … … …
81.2789449689422,	−0.0048846415593919446,	110.38050475265473,
−0.22137416866235676,	48.551378225507776,	63.03698835482671,
35.7418841844737,	83.53473079490693,	−0.02281540057297471,
109.66348320903144,	−0.21658630841811755	50.861356166273346,
58.92089725679235,	43.81047527717123,	… … … … … … … … …
72.21634873205072,	17.239506104410605,	103.89155079639312,
−0.17682477240153638,	68.98820498572555,	23.801081041461266,
98.12493764377616,	−0.13510761998171994,	85.34225448761212,
−0.037108830233576584,	108.43172291299035,	−0.20828105984144446,
54.811095439651815,	51.64760626842221,	57.49569318219606,
46.55590588151394,	66.50143584533328,	28.821566838437924,
92.59760077348993,	−0.09352520129650865,	… … … … … … … … …
62.574095563374875,	36.65846965189703,	82.21852655174172,
−0.01236406650119004,	110.1940684700879,	−0.22013259166617039,
49.15259798562827,	61.97624638479911,	37.839095552679645

Step - 3: Perform the Exclusive-OR (XOR) operation for sequences produced by Step 2 with the pixels of the original image. (After converting above decimal values into binary)

1101111000100000482,	1101111000099363488,	1101111000100110203,
1101111000099000650,	1101111000100100110,	1101111000099364504,
1101111000100100202,	1101111000099001656,	… … … … … … … … …
1101111000100100110,	1101111000100009407,	1101111000100006387,
1101111000100003595		

Coupled Lattice Map

Step - 1: The chaotic sequences $x_1, x_2 = \{x_1, x_2, x_3, \dots, x_m\}$ are produced with the length of m, and $y_1, y_2 = \{y_1, y_2, y_3, \dots, y_n\}$ with the length of n similar to CML chaos mapping.

−1.1227856299999996,	−7.9707143571985934,	−739129.3279655317,
−3097589966812.097,	−5.440401062614647e+25	−1.6782045430429897e+52,
−361.05091198417364	−1.5968820668605035e+105,	−1.445868334206202e+211
And So On……		
−5.382397044,	−14.064056390363792,	−24.90977559867699,
−38.458985542608005,	−55.38557995084281,	… … … … … … … … …
−951.7829429777029,	… … … … … … … … …	… … … … … … … … …

−104198709.07980004,	−130172120.23470365,	−162619871.64136776,
−203155809.14566824,	−253796058.71979067,	−317059302.0247555,
−396092047.4618622,	−494825127.2885856,	−618169204.4575567,
−772259013.0542831,	… … … … … … … …	−1.0696724691213476e+25
… … … … … … … …	−1.3363075861542877e+25	−1.669406305539795e+25,
−2.0855358765090883e+25,	−2.6053932333747555e+25,	−3.2548343938716144e+25,
−4.066160453563204e+25,	−5.079724137501996e+25,	−6.345936813808843e+25,
−7.927775791513346e+25,	−9.903916607512294e+25	

Step - 2: x, y chaotic sequences are arranged in rising sequences, producing position sequences w_1, w_2.

List in rising order:

−4.0482329554222246e+189,	−3.3212358404609667e+94,	−9.512985303565721e+46,
−9.903916607512294e+25,	−7.927775791513346e+25,	−6.345936813808843e+25,
−1.8030422060086812e+24,	−1.4432789489594556e+24,	… … … … … … … …
−8.063121201687744e+19,	−6.454276585718791e+19,	−5.166446739785852e+19,
−4.135579187000589e+19,	−3.3104019209653883e+19,	−2.6498733025783005e+19,
… … … … … … … …	−135.9499791899694,	−102.94826985880484
−76.53143269402949,	−76.53143269402949,	−8.205787630000001

Step - 3: In the last step, the pixel confusion is performed, using w_2, w_3 as the row, and column sequences of the data matrix R.

$$R(i,1) = R(w_1(i, w_2(j))).$$

```
[[[255  255  255                                  [12   72   14
 255  255  255          ……..                     109  251  211
 234  203   97]                                   255  255  255]]

 [[255  255  255                                  [16  199  161
 255  255  255          ……..                      38   76  222
 135   36   25]                                   255  255  255]]

 [[255  255  255                                  [22  148   94
 255  255  255          ……..                     107   54   85
 210  179  222]                                   255  255  255]]

                         ……..
                         ……..

 [[255  255  255                                  [255  255  255
 255  255  255          ……..                     255  255  255
 255  255  255]                                   255  255  255]]

 [[255  255  255                                  [[255  255  255
 255  255  255          ……..                     255  255  255
 255  255  255]                                   255  255  255]]

 [[255  255  255                                  [255  255  255
 255  255  255          ……..                     255  255  255
 255  255  255]                                   255  255  255]]]
```

Algorithm

Step - 1: A image G is arranged. The image's size is set to $m \times n$. Additionally, a data matrix R is placed. By evaluating two gaussian maps, a chaotic sequence is generated. Making XOR with the primary image, the diffusion is completed.

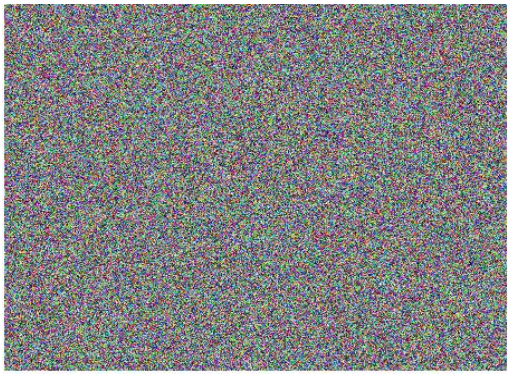

Figure 2: Diffused Image

Step - 2: For the diffused image in step1, the Gabor decomposition is performed and Gabor coefficient is extracted.

Array = $[91, 77, 86, \ldots, 125, 132, 158]$, $[79, 71, 80, \ldots, 112, 144, 168]$,
$[80, 82, 89, \ldots, 79, 159, 161]$, … … … … … … … … …
$[42, 20, 14, \ldots \ldots, 96, 81, 81]$, $[36, 22, 20, \ldots, 101, 92, 92]$,
$[52, 43, 30, \ldots, 113, 107, 102]$

dtype = uint8

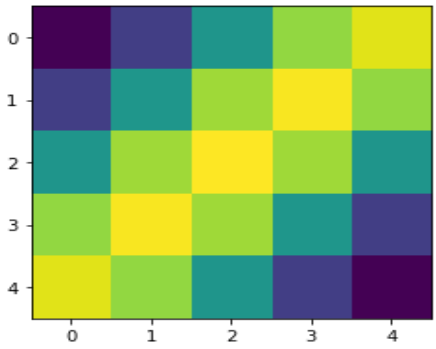

Figure 3: Image from Gabor Coefficient

Step - 3: Utilizing the CML, the chaotic sequence is produced, established in step 2, the position confusion is performed.

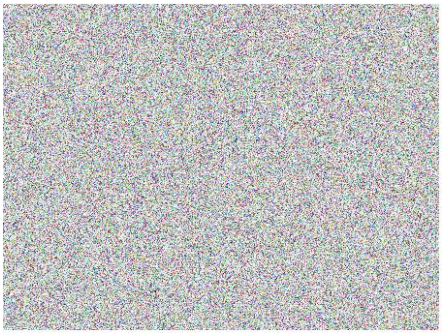

Figure 4: Confused Image

Step - 4: In the last step, the confused image can be reconstructed by Gabor. Thus, the encrypted image is obtained.

[[1 0 1 1 1 0 0 0 1 0], [0 1 1 1 0 0 1 0 0 0]]

Figure 5: Encrypted Image

VI. Results And Discussion

Image quality and vision outcomes were generated as a result of the experiment. The following parameters are used to evaluate image quality:

Number of Pixels Change Ratio (NPCR)

When the difference between two encrypted images is negligible, NPCRs are used to verify the number of changing pixels between them. The optimal NPCR value is 99.68%. The NPCR can be mathematically defined as follows:

$$\text{NPCR} = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{D(i,j)}{M \times N}\ 100\%$$

Where $D(i,j) = \begin{cases} 0; & if\ C_1(i,j) = C_2(i,j) \\ 1; & if\ C_1(i,j) \neq C_2(i,j) \end{cases}$

m, n is the weight and height of the encrypted interferogram,
$C_1(i,j)$ is the interferogram encrypted before pixel change,
$C_2(i,j)$ is the interferogram encrypted after pixel change,
$D(i,j)$ is the bipolar network

Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR)

The PSNR block analyses the peak signal-to-noise ratio between two images in decibels. The PSNR ratio is used to compare the quality of the original and encrypted images. The better the quality of the compressed or reconstructed image, the higher the PSNR.

To compare image compression quality, the mean square error (MSE) and peak signal-to-noise ratio (PSNR) are evaluated. The MSE is a measure of the peak error between the encrypted and original image, whereas the PSNR is a measure of the cumulative squared error.

The smaller the MSE value, the smaller the error.

The PSNR is calculated by first calculating the mean-squared error (MSE) using the equation:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - k(i,j)]^2$$
$$= \sum_{i,j} \frac{[I(i,j) - k(i,j)]^2}{mn}$$

PSNR can be calculated as:

$$\text{PSNR} = 20\ \log_{10} \frac{256}{\sqrt{MSE}}$$

The optimum MSE value is 2.16 and the optimum PSNR is 41.47% for this image.

Unified Average Changing Intensity (UACI)

UACI is used to calculate the average intensity of the difference between the two encrypted images (C_1 and C_2). It is used to determine the strength of an encryption scheme. Its quality is determined by the format and size of the image. The average intensity variation between the ciphered and original images is measured using UACI. The highest UACI suggests that the recommended technique is resistant enough to a variety of attacks.

For an image of size $m \times n$, UACI is calculated as follows:

$$\text{UACI} = \frac{1}{mn} \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{256}\ 100\%$$

The optimum UACI value is 33.48% for this image.

Time taken for Encryption and Decryption of an image

The Encryption took 0.0sec of time and the decryption took 0.734375sec of time when calculated through PYTHON.

Visual Testing

The visual testing is done online on https://www.textcompare.org/image/.

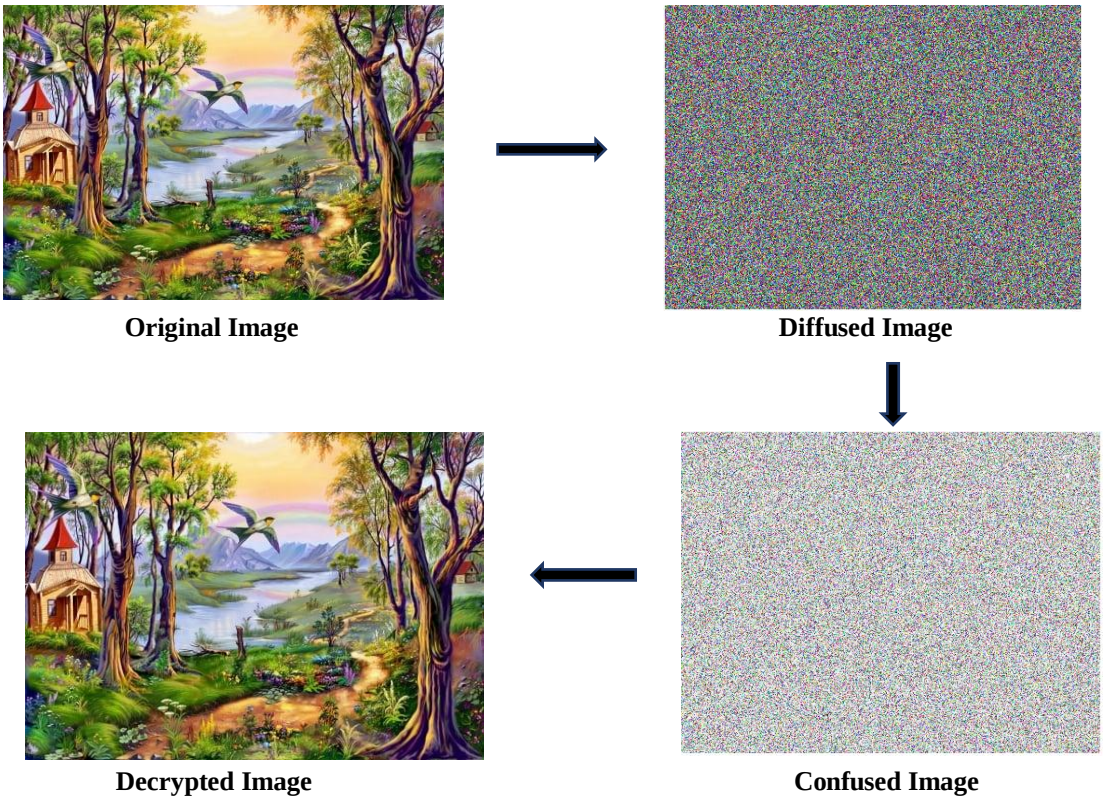

Figure 6: Visual Testing

By comparing original and encrypted images, the encrypted image is 99.20% different compared to the original image. The white dot indicates the similar pixel value of original image and encrypted image. The difference is found with full transparency when α is ignored alongwith scale to same size and movement with different intensity.

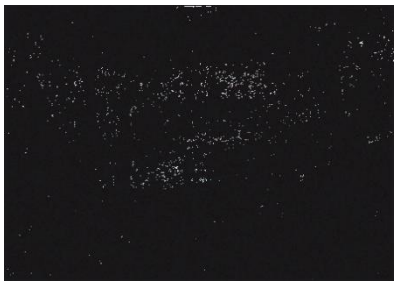

Figure7:Difference in image when it is scale to same size

By comparing original and encrypted images, the encrypted image is 99.32% different compared to the original image. The white dot indicates the similar pixel value of original image and encrypted image. The difference is found with full transparency when α is ignored alongwith original size and movement with different intensity.

Figure8:Difference in image when it is of original size

VII. Conclusion

Inspired from the latest trend of various chaos based image cryptosystems studies, the present work is carried out. The current work deals with a chaotic-based algorithm using characteristics of the chaotic map and Gabor transform. For the encryption process, the image diffusion operation is executed. Moreover, by performing the Gabor transformation, the calculation amount in confusion was slightly reduced by hyper-chaotic sequences. The simulation outcomes with the standard metrics show that the proposed algorithm has a high dependence on keys. This algorithm includes a decent encryption outcome.

In addition, the encryption performance analysis criteria such as PSNR, NPCR, UACI and MSE are recorded.

On the basis of this analysis, Gabor transformation gives a high encryption effect. It can resist noise and some attacks too. Also, it possesses higher degree of robustness and normalized correlation. Input attack does not affect the image encryption and decryption. It is more efficient than the wavelet transformation. The

References

[1]. Enayatifar, R., Abdullah, A. H., & Isnin, I. F. (2014). Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Optics and Lasers in Engineering, 56, 83-93.
[2]. Hafsa, A.; Gafsi, M.; Malek, J.; Machhout, M. FPGA Implementation of Improved Security Approach for Medical Image Encryption and Decryption. Sci. Program. 2021, 6610655.
[3]. Li, C. L., Li, H. M., Li, F. D., Wei, D. Q., Yang, X. B., & Zhang, J. (2018). Multiple-image encryption by using a robust chaotic map in the wavelet transform domain. Optik, 171, 277-286.
[4]. Patidar, V.; Pareek, N.; Purohit, G.; Sud, K. A robust and secure chaotic standard map based pseudorandom permutation substitution scheme for image encryption. Opt. Commun. 2011, 284, pp-4331–4339.
[5]. Pourasad, Y., Ranjbarzadeh, R., & Mardani, A. (2021). A new algorithm for digital image encryption based on chaos theory. Entropy, 23(3), 341.
[6]. Satish, T.J.; Theja, M.N.S.; Kumar, G.G.; Thanikaiselvan, V. Image Encryption Using Integer Wavelet Transform, Logistic Map and XOR Encryption. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp-704–709.
[7]. Xu, Y.; Wang, H.; Li, Y.; Pei, B. Image encryption based on synchronization of fractional chaotic systems. Commun. Nonlinear Sci. Numer. Simul. 2014, 19, pp-3735–3744.
[8]. Sarmah, H. K., Das, M. C., Baishya, T. K., & Paul, R. (2016). Characterising chaos in a gaussian map. Int J Adv Sci Tech Res, 6(1), 160-172.
[9]. https://www.textcompare.org/image/

Srushti Gandhi. "Digital Image Encryption based on Gauss Map and Gabor Transformation." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 24(3), 2022, pp. 49-59.