Multimedia Tools and Applications (2024) 83:43761–43776
https://doi.org/10.1007/s11042-024-18634-w

Accelerating randomized image secret sharing with GPU: contrast enhancement and secure reconstruction using progressive and convolutional approaches

Raviraja Holla M[1] · Suma D[2] · Alwyn R. Pais[3]

Received: 24 May 2023 / Revised: 31 January 2024 / Accepted: 12 February 2024 /
Published online: 2 March 2024

Abstract

Image Secret Sharing (ISS) is a cryptographic technique used to distribute secret images among multiple users. However, current Visual Secret Sharing (VSS) schemes produce a halftone image with only 50% contrast when reconstructing the original image. To overcome this limitation, the Randomized Image Secret Sharing (RISS) scheme was introduced. RISS achieves a higher contrast of 70% when extracting the secret image but comes with a high computational cost. This research paper presents a novel approach called Graphics Processing Unit (GPU)-based Randomized Image Secret Sharing (GRISS), which utilizes data parallelism within the RISS pipeline. The proposed technique also incorporates an Autoencoder-based Single Image Super-Resolution (ASISR) to enhance the contrast of the recovered image. The performance of GRISS is evaluated against RISS, and the contrast of the ASISR images is compared to current benchmark models. The results demonstrate that GRISS outperforms state-of-the-art models in both efficiency and effectiveness.

Keywords Gpu · Shares · Autoencoder · Super resolution

1 Introduction

Image Secret Sharing (ISS) is a type of image cryptography used to secure secret visual information using encrypted shares. There are several visual cryptography techniques available

✉ Raviraja Holla M
raviraj.holla@manipal.edu

Suma D
suma.d@manipal.edu

[1] Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal 576104, India

[2] Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal, Karnataka 576104, India

[3] Information Security Research Laboratory Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal 575025, India

for securely transmitting visual media [1–4]. However, most ISS techniques suffer from common problems, such as poor contrast when recovering secret images. The maximum achievable contrast with ISS techniques is typically 50% [5]. Existing VSS schemes typically convert grayscale and color images into monotone images using halftone techniques, resulting in recovered images that are always monotone. To address these limitations, the Block-based Progressive Image Secret Sharing (BPISS) scheme was proposed by Hou et al. [6]. This scheme achieves a contrast of 50% but recovers secret images as halftone images. Kim et al. [7] proposed an efficient method for high-capacity separable reversible data hiding in encrypted halftoned images using Hamming code, ensuring a large payload with low distortion. Experimental results confirm its effectiveness in information embedding and image reconstruction.

In an effort to overcome these limitations, the Randomized Image Secret Sharing (RISS) scheme was introduced by Mhala et al. [5]. RISS utilizes a data embedding technique to recover secret images with higher contrast, ranging from 70-80% for noise-like shares to 70-90% for meaningful shares. Enhancing RISS in two dimensions makes it more suitable for time-sensitive and health-critical applications. Utilizing parallelism in the RISS pipeline decreases computational costs [8, 9]. Improving the contrast of RISS improves the image quality [10].

Recent studies in the field of parallel computing have highlighted the benefits of using Graphics Processing Unit (GPU) and Compute Unified Device Architecture (CUDA) for faster computation [11]. To address these concerns, this research paper proposes an efficient GPU-based Randomized Image Secret Sharing (GRISS) scheme. GRISS utilizes data parallelism within the RISS pipeline and incorporates the Autoencoder-based Single Image Super-Resolution (ASISR) technique to enhance the contrast of recovered images. Single Image Super-Resolution (SISR) methods rest on two fundamental principles: precision and robustness. Autoencoders have demonstrated their capability to provide reconstructions that are both accurate and stable [12]. Nonlinear autoencoders demonstrate the capability to capture and represent the influence of each latent variable on super-resolution reconstruction through their ability to model nonlinear modes in the weight distribution across the latent space representation and the entire field [13]. An autoencoder is trained to encode and decode data, serving the purposes of dimensionality reduction and feature learning. This is accomplished through various regularized and variational encoders [14].

The performance of GRISS is compared against the RISS scheme, demonstrating improved speed and contrast. The proposed GRISS scheme outperforms state-of-the-art models in both efficiency and effectiveness. This system adopts a user-centric design, prioritizing simplicity in deployment and user interaction. It operates by generating a number of digital shares corresponding to the participants, and each share is distributed to the respective participant. Notably, individual shares contain no discernible information about the secret. The secret image is reconstructed only when participants collectively combine their digital shares. Subsequently, by feeding this reconstructed secret image into the autoencoder, a superior-quality digital image is obtained. This process ensures the security of the shared secret while delivering a high-fidelity reconstructed image through the integration of advanced autoencoder techniques. Resource optimization strategies are implemented to mitigate concerns about higher operational costs and resource requirements associated with GPU utilization. The GRISS system effectively harnesses GPU resources through parallel processing, ensuring efficient operation even in environments with constrained computational resources. Moreover, the integration of ASISR into the GRISS pipeline stands as a testament to the system's commitment to delivering high-quality results. This strategic integration not only elevates the visual quality of the secret images but also maintains computational efficiency through optimized GPU utilization.

2 Related works

Image Secret Sharing (ISS) is a cryptographic technique that involves dividing an image into multiple shares, which are distributed among a group of participants. The shares are constructed in such a way that only a specific threshold of shares is required to reconstruct the original image, while any fewer shares reveal no information about the image. Based on the recovery of the secret image, the ISS is categorized into Random-Grid ISS, Traditional ISS schemes, and Progressive ISS.

The basic Random-Grid ISS uses two random-grids [15]. In this approach, An encoded random-grid is obtained using a master random-grid and an input image. The secret image gets revealed when these two random-grids overlaid. The basic RG ISS is continuously evolving from its extension to the colour images for enhancing security, efficiency, and practicality [10, 16–18]. Traditional ISS recover the secret image with a minimal k shares out of n shares when stacked together. The term Visual Cryptography (VC) was founded by Naor and Shamir [19]. Basis matrices are used to veil the original pixels of the secret images. Generating uniform shares for each participant and the pixel expansion are the drawbacks of this model [20].

Progressive ISS works like a jig-saw puzzle [6]. Block-based progressive Image Secret Sharing schemes are a variation of Image Secret Sharing techniques that operate at the block level, dividing the image into smaller blocks and generating shares for each block individually. This approach provides progressive reconstruction, meaning that the original image can be partially reconstructed using a subset of the shares, and the reconstruction quality improves as more shares are combined. Progressive ISS schemes offer advantages such as efficient storage and transmission of shares, progressive visualization of the secret image, and flexibility in handling different block sizes or resolutions. However, they also introduce challenges related to block misalignment, visual artifacts at block boundaries, and increased computational requirements for generating and combining shares. Also, progressive recovery approaches attained utmost 50% recovered image contrast [5].

To overcome the issues of progressive ISS, a novel Randomized Image Secret Sharing (RISS) scheme is invented by Mhala et al. [5]. This technique combines BPVSS for generating secret shares and reversible Discrete Cosine Transformation (DCT) data embedding technique. The original inventors of RISS made advancements to enhance the contrast of the recovered images, termed the contrast-enhanced progressive ISS (CEPISS) scheme [21]. Kang et al. introduced a novel image-based data hiding method utilizing the Shamir threshold scheme, which disassembles secret messages into subsecrets embedded in smooth blocks of different images, effectively avoiding security issues faced by dual-image and multi-image data hiding methods [22]. Mhala et al. presented a secure ISS (CNN-SISS) scheme using convolutional neural networks (CNN) [23]. These developments aimed to improve the contrast quality of the reconstructed images obtained through the RISS framework.

This study makes a notable contribution by introducing a novel approach, the GPU-based Randomized Image Secret Sharing (GRISS) system, aimed at enhancing the efficiency of Randomized Image Secret Sharing (RISS) through advanced data-parallel methods. This contribution is underscored by the achieved significant speedup improvements in both sender and receiver pipelines through the utilization of GPU-based parallel processing. The method effectively harnesses GPU constant memory for storing basis matrices, optimizing broadcasting and computation. The substantial reduction in execution time, especially with larger image sizes and a higher number of shares, demonstrates a notable twofold speedup improvement compared to the traditional RISS method. Moreover, the study addresses the crucial aspect of image quality in Image Secret Sharing (ISS) systems by employing a state-of-the-art autoencoder designed for single image super resolution. Experimental results confirm that the GRISS method successfully reconstructs the secured image while preserving the quality of the original secret image, thereby significantly enhancing the overall value and effectiveness of ISS systems in real-time security applications.

3 Proposed GRISS methodology

This section aims to provide a comprehensive explanation of the parallel methods illustrated in Fig. 1. These methods employ two distinct modules: the Sender and the Receiver. Each module is explained separately in the following sections.

3.1 Sender module

The Sender module comprises two primary stages: share generation and embedding of Low Resolution (LR) secret image information into the shares. To provide further details on these stages, we introduce Algorithms 3 and 4. In Algorithm 3, basis matrices generated using GPU, as outlined in Algorithm 1, are utilized in conjunction with the secret image during the share generation process. Additionally, Algorithm 2 employs GPU processing to generate the halftone equivalent of a grayscale or color image, which is subsequently utilized in the share generation process performed by Algorithm 3.

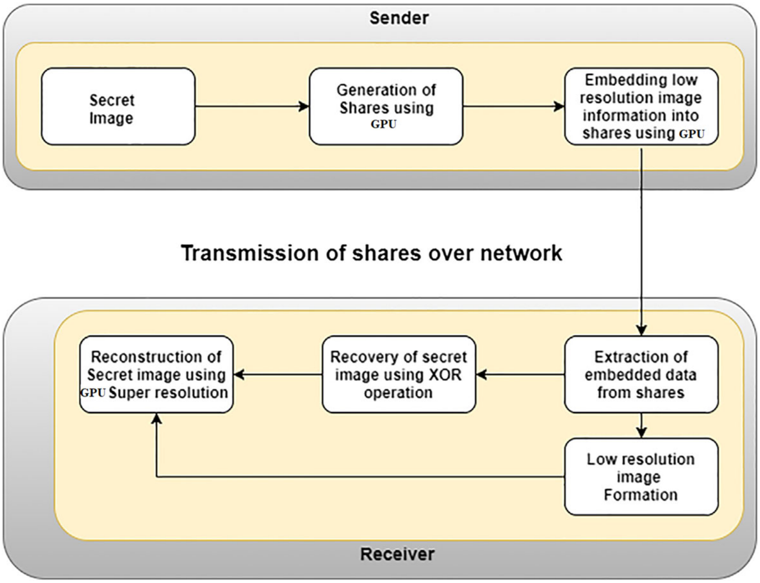

Sender
Secret Image
Generation of Shares using GPU
Embedding low resolution image information into shares using GPU
Transmission of shares over network
Reconstruction of Secret image using GPU Super resolution
Recovery of secret image using XOR operation
Extraction of embedded data from shares
Low resolution image Formation
Receiver

Fig. 1 Proposed methodology

Algorithm 1 uses GPU to generate $n + 1$ basis matrices of two rows and n columns for creating n shares, where the first row of $B0$ contains all zeros and the second row contains all ones, and the remaining basis matrices follow the pattern of having only one column with 1 in the first row and the second row being the complement of the first row. Algorithm 4 is a GPU-based parallel computation to perform an 8×8 Discrete Cosine Transform (DCT) of the n shares, quantize those blocks, find ceaseless zeros, and embedding the original image information in the mid-frequency area of the block.

Algorithm 1 Generating basis matrices using GPU.

```
1 Start by initializing an empty list to store the basis matrices.
2 Create a GPU kernel function generateBasisMatrices:

  – Calculate the global index of the current thread.
  – If the global index is 0:
      – Generate B0:
          • Set the corresponding elements of the first row to 0 and the second row to 1.
  – If the global index is between 1 and n (inclusive):
      – Generate B1 to Bn:
          • Set the corresponding element in the first row to 1 (only one column contains 1).
          • Set the corresponding element in the second row to the ones complement of the first row.
3 Launch the GPU kernel generateBasisMatrices, with the desired number of blocks and threads per block.
4 End
```

Algorithm 2 Generating halftone image using GPU.

```
1 Load the RGB image into device memory.
2 Allocate memory for the output colored halftone image.
3 Calculate the thread indices (x, y) using blockIdx and threadIdx.
4 Calculate the corresponding pixel position in the output image based on the thread indices.
5 Fetch the RGB values from the input image for the corresponding pixel position. 6 Convert the RGB values to grayscale using a luminosity method.
7 Calculate the halftone dot pattern based on the grayscale value.
8 Set the RGB values of the output image pixels according to the halftone dot pattern and the original RGB values.
9 Write the resulting colored halftone image to global memory.
10 stop
```

3.2 Receiver module

The Receiver module is designed as the inverse of the Sender module. Its main function is to extract the embedded data, which consists of the low-resolution (LR) secret image used for training a single image super-resolution autoencoder. Additionally, the Receiver module generates a secret image by performing an XOR operation on the shares without the embedded information.

The emphasis in this context lies on utilizing a state-of-the-art autoencoder for image super-resolution, which takes advantage of GPU acceleration. This allows for efficient and high-quality enhancement of the LR input, resulting in a secret image with enhanced contrast.

Algorithm 3 Generating shares using GPU.

```
1 Load the basis matrices B0 to Bn from constant memory.
2 Calculate the number of blocks needed to cover the image and distribute them across the GPU thread blocks.
3 In the GPU kernel function:
    – Calculate the global index of the current thread.
    – Determine the block index and pixel index within the block
    – Determine the corresponding basis matrix index based on the color of the pixel.
    – If the pixel is black (0), randomly choose a row from B0 and assign the successive column values to the corresponding shares in the block.
    – If the pixel is white (1), randomly choose a row from B1 to Bn and update the corresponding blocks in the shares with the successive column value
4 Synchronize the threads within each block.
5 Return the generated shares.
```

Algorithm 4 Data embedding using GPU.

```
1 For each share in GPU memory, do in parallel :
    a Compute the DCT of the share block using the 8 × 8 GPU DCT library function and store the result in GPU memory.
    b Quantize the DCT coefficients to reduce the number of bits required for storage.
    c Find the location of ceaseless zeros in the quantized DCT coefficients.
    d Embed the original image information in the mid-frequency area of the block by replacing the ceaseless zeros with the corresponding binary values of the original image pixels.
    e Store the resulting block in GPU memory.
2 Return embedded shares
```

Image resolution pertains to the level of detail present in an image, and in various applications, there is a strong preference for high-resolution (HR) images compared to low-resolution (LR) counterparts. This preference stems from the appeal HR images hold for both machine analysis and human perception. To achieve higher resolution, super-resolution (SR) techniques are employed, which aim to extract HR images from their LR observations while preserving intricate and non-redundant details.

The process of super-resolution involves solving a set of linear equations, as represented by (2) and (3). Equation (1) introduces the SR model used to obtain these equations. In this model, the desired HR image is denoted by H, and L_j represents the j^{th} LR sample out of J samples obtained from H. Additionally, the model incorporates various components such as F_j for capturing motion terms, B_j to account for blurring effects, D_j as the down-sampling operator, and V_j to represent noise. By carefully manipulating these elements, the SR model endeavors to generate HR images that exhibit enhanced resolution and maintain the fidelity of fine details. The estimation of sparse matrices D_j, B_j, F_j, and M is carried out based on LR samples.

$$L_j = D_j B_j F_j H + V_j \qquad for \qquad 1 \leq j \leq J \tag{1}$$

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_j \end{bmatrix} = \begin{bmatrix} D_1 B_1 F_1 \\ \vdots \\ D_j B_j F_j \end{bmatrix} H + \underline{V} \tag{2}$$

$$\underline{\mathbf{Y}} = \mathbf{MH} + \underline{\mathbf{V}} \tag{3}$$

Figure 2 illustrates a schematic representation of the proposed autoencoder, and a summary is presented in Table 1. The representation of input and output layer corresponds to a vector of

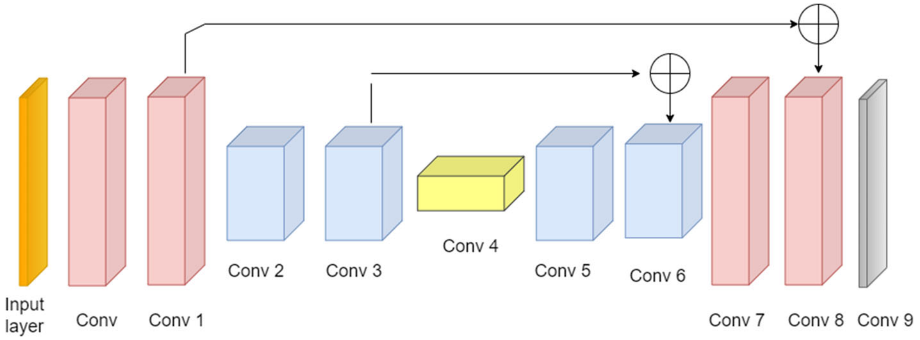


Fig. 2 Schematic of an autoencoder

values that pertain to a time series at a particular pixel location. Similarly, the hidden layers also represent temporal vectors, but they encompass more general patterns of variability instead of being specific to a pixel. Every connection in the network is assigned a random weight initially and is adjusted during training to ensure that the output layer closely matches the input layer. This adjustment is made while utilizing only the latent patterns in the hidden layer as input. The encoder component of the network develops the hidden layer, while the decoder component develops the output layer.

Autoencoder neural networks have the capability to learn LR samples without the need for explicit labels. In this context, let $s \in R^N$ represent the original secret image, and $s' \in R^N$ denote the restored distorted image, where $s' = Ms$. Here, $M \in R^{N\times N}$ is assumed to be

Table 1 Autoencoder summary

Autoencoder layers	Output shape	Number of parameters
Input layer	(None, 80, 80, 3)	0
Conv (2D)	(None, 80, 80, 64)	1792
Conv_1 (2D)	(None, 80, 80, 64)	36928
Max pooling (2D)	(None, 40, 40, 64)	0
Conv_2 (2D)	(None, 40, 40, 128)	73856
Conv_3 (2D)	(None, 40, 40, 128)	147584
Max pooling_1 (2D)	(None, 20, 20, 128)	0
Conv_4 (2D)	(None, 20, 20, 256)	295168
Up sampling (2D)	(None, 40, 40, 256)	0
Conv_5 (2D)	(None, 40, 40, 128)	295040
Conv_6 (2D)	(None, 40, 40, 128)	147584
Add (Conv_3, Conv_6)	(None, 40, 40, 128)	0
Up sampling_1 (2D)	(None, 80, 80, 128)	0
Conv_7 (2D)	(None, 80, 80, 64)	73792
Conv_8 (2D)	(None, 80, 80, 64)	36928
Add_1(Conv_1, Conv_8)	(None, 80, 80, 64)	0
Conv_9 (2D)	(None, 80, 80, 3)	1731

sparaTrainable parameters = Total number of parameters = 1,110,403
Non-trainable parameters = 0

a high-dimensional matrix responsible for the distortion applied to the secret image. The elements of M can be characterized by the activation function $h(s') \in R^K$ in (4) and the reconstruction function $\hat{s}(s') \in R^N$ in (5).

$$h(s') = \sigma(Ws' + b) \tag{4}$$

$$\hat{s}(s') = \sigma'(W'h + b') \tag{5}$$

These functions are learned by the autoencoder model, where σ represents the encoding function and σ' represents the decoding function. The sigmoid function, defined in (6), is typically used for a single autoencoder layer consisting of K units. The weights for each encoder layer are denoted as $W \in R^{K \times N}$, and the corresponding biases are represented by $b \in R^K$. Similarly, the weights for each decoder layer are denoted as $W' \in R^{N \times K}$, and the biases are represented by $b' \in R^K$.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

4 Experimental results

The image dataset used for the experiments can be found at [24, 25]. This dataset comprises a total of 44 images, consisting of 16 color images, 28 grayscale images, and two binary images. Additionally, a publicly available high-dimensional medical image dataset related to breast cancer was employed, which can be accessed at [24]. Figure 3 visually depicts the sequential image transformations involved in the GRISS process, showcasing the step-by-step progression from the sender to the receiver during the reconstruction of the secure image.

The forthcoming sections will examine the performance of the proposed GRISS from multiple perspectives. Firstly, the execution time efficiency will be assessed, followed by an evaluation of the quality of reconstructed images using objective quality assessment parameters. Additionally, a sensitivity analysis of the autoencoder parameters will be conducted.

4.1 Execution time analysis

The execution time comparison between sequential RISS and parallel GRISS for grayscale images with different sizes, using 4 shares, is depicted in Fig. 4. Similarly, Fig. 5 illustrates the comparison for 8 shares. For color images, the execution time comparisons can be seen in Figs. 5, 6 and 7. As the number of shares and image size increase from grayscale to color images, GRISS demonstrates efficient scalability compared to RISS. This is evident from the significant increase in the speedup, which represents the ratio of execution time on RISS to GRISS.

4.2 Objective image quality analysis

Assuming S' represents the distorted image and S represents the reference plain image, where M is the width of S and N is the height of S, both images S and S' are of the same size, denoted by (7) and (8). In the equations, $s(i, j)$ and $s'(i, j)$ represent the pixel intensities at

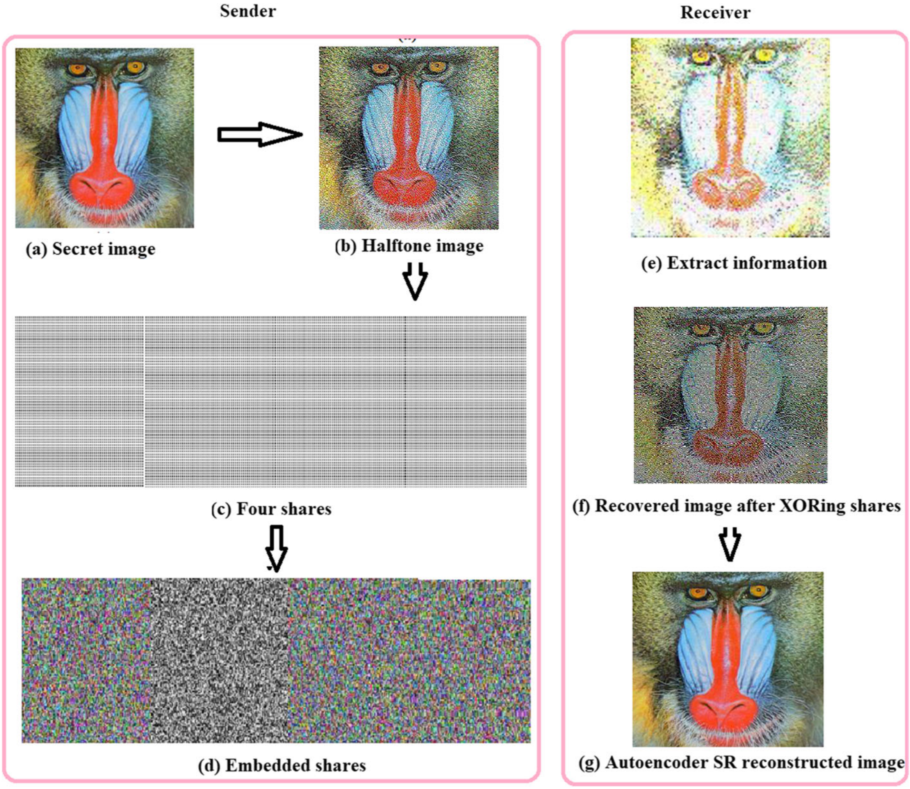


Fig. 3 Illustration of secret image transformation from sender to receiver

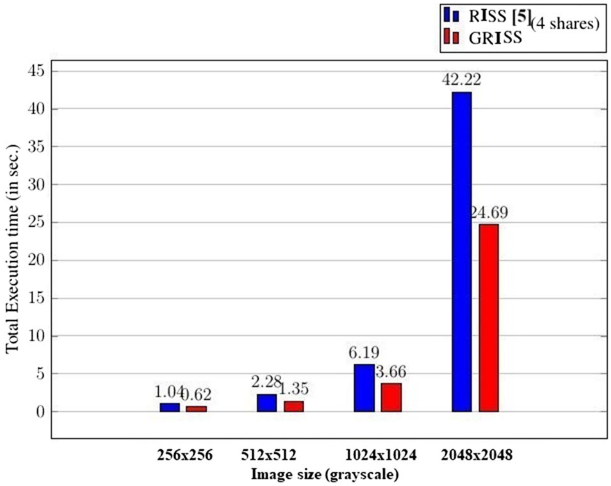


Fig. 4 Execution time comparison: RISS [5] vs. GRISS on a **4**-share **grayscale** image

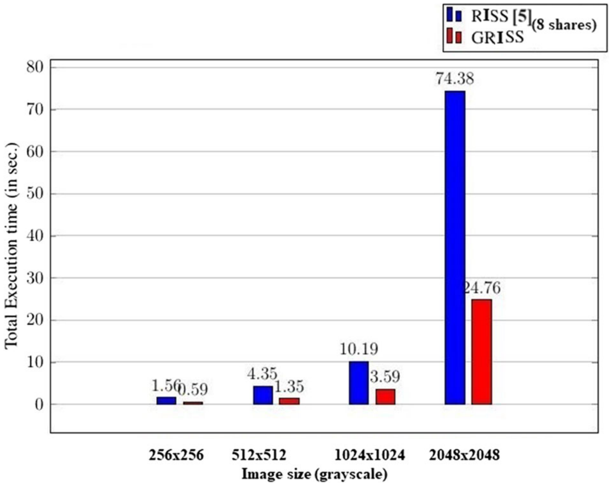


Fig. 5 Execution time comparison: RISS [5] vs. GRISS on a **8**-share **grayscale** image

the position (i, j) in images S and S', respectively.

$$S = \{s(i, j)|(0 \le i \le M - 1), (0 \le j \le N - 1)\} \tag{7}$$

$$S' = \left\{s'(i, j)|(0 \le i \le M - 1), (0 \le j \le N - 1)\right\} \tag{8}$$

4.2.1 Mean Squared Error (*MSE*)

The Mean Squared Error (MSE) is a metric that calculates the average of the squared differences between corresponding intensities. As the difference between intensities approaches

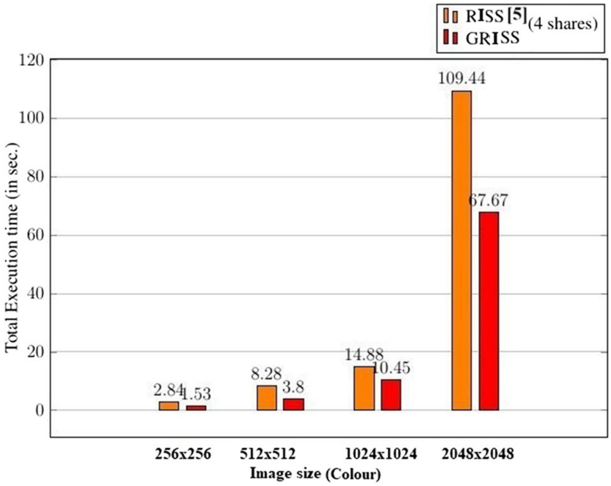


Fig. 6 Execution time comparison: RISS [5] vs. GRISS on a **4**-share **colour** image

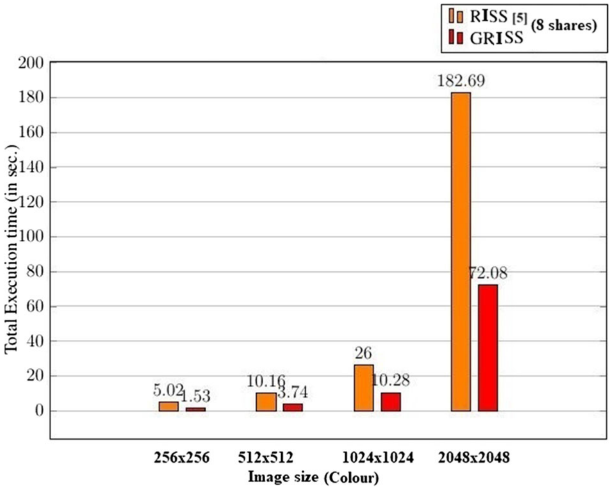


Fig. 7 Execution time comparison: RISS [5] vs. GRISS on a **8**-share **colour** image

zero, the MSE decreases. Equation (9) provides the definition of the MSE.

$$MSE(S, S') = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [s(i, j) - s'(i, j)]^2 \tag{9}$$

Based on the data presented in Table 2, it is evident that the proposed model achieves the lowest MSE value of 0.0017 when compared to other benchmark models.

4.2.2 Normalized Cross Correlation (*NCC*)

NCC is another metric to demonstrate similarity between the source and target images. The high value of NCC implies the more similarity amongst the source and target images. Equation (10) defines the NCC.

$$NCC(S, S') = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [\frac{s(i, j) \times s'(i, j)}{s^2\ (i, j)}] \tag{10}$$

Table 2 gives the comparison of NCC values of the proposed model over the other models. In Table 2, the proposed model has a maximum of 99.03% correlation when compared to the other benchmark models.

Table 2 MSE, NCC, NAE, and $SSIM$ values of the proposed model

Model⇒ Parameters ⇓	RISS [5]	CEPISS [21]	CNN-SISS [23]	Proposed GRISS
MSE	0.107	0.035	0.014	0.0017
NCC	0.6692	0.9876	0.9888	0.9903
NAE	0.2776	0.0413	0.0380	0.0238
SSIM	0.2750	0.8740	0.8967	0.9826

4.2.3 Normalized Absolute Error (*NAE*)

The dissimilarity between the source and target images can also be quantified using NAE parameter, as defined in (11).

$$NAE(S, S') = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [\frac{s(i, j) - s'(i, j)}{s(i, j)}] \tag{11}$$

A lower NAE value indicates better visual quality of the recovered target image. From the data presented in Table 2, it is apparent that the proposed model surpasses the comparative models, achieving an impressive NAE value of 0.0238.

4.2.4 Structural Similarity Index (*SSIM*)

The Structural Similarity Index ($SSIM$), as described in (12), is a metric that ranges between -1 and 1 in the spatial domain. This metric compares the contrast, luminance, and format of the source and target images [26].

$$SSIM(S, S') = [l(S, S')]^{\alpha}.[c(S, S')]^{\beta}.[s(S, S')]^{\gamma} \tag{12}$$

Where,

$$\begin{aligned} l(S, S') &= \frac{2\mu_S \mu_{S'} + K1}{\mu_S^2 + \mu_{S'}^2 + K2} \\ c(S, S') &= \frac{2\sigma_S \sigma_{S'} + K2}{\sigma_S^2 + \sigma_{S'}^2 + K2} \\ s(S, S') &= \frac{\sigma_{S'} + K3}{\sigma_S \sigma_{S'} + K3} \end{aligned}$$

When two images are similar, their $SSIM$ value will be close to 1. On the other hand, for distinctly different images, the $SSIM$ value decreases, reaching a minimum of -1. Equation (12) can be simplified to (13) under the assumption that α, β, and γ are all equal to 1, and $K3$ is equal to $K2/2$.

$$SSIM(S, S') = \frac{(2\mu_S \mu_{S'} + K1)(2\sigma_{SS'} + K2)}{(\mu_S^2 + \mu_{S'}^2 + K1)(\sigma_S^2 + \sigma_{S'}^2 + K2)} \tag{13}$$

In these simplified equations, σ_S and $\sigma_{S'}$ represent the standard deviations, μ_S and $\mu_{S'}$ represent the means, and $\sigma_{SS'}$ represents the cross-covariance between S and S'.

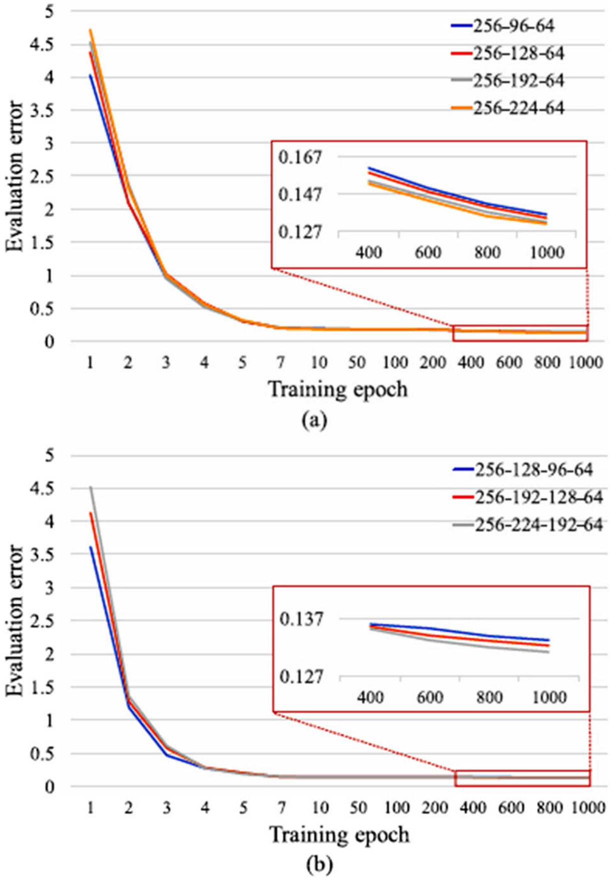
In Table 2, the $SSIM$ values of the proposed model for the compared images S and S' are provided, along with the values for other reference models. Notably, the presented model achieved a significantly higher $SSIM$ value of 98.26%, surpassing the state-of-the-art models.

4.3 Autoencoder sensitivity analysis

In this section, we conducted additional tests to evaluate the sensitivity of the method's parameters, specifically focusing on the number of hidden layers and neurons in the autoencoder, using a scale factor of 3. The implemented autoencoder model consists of a single hidden layer with 192 neurons. For simplicity, we refer to this architecture as 256-192-64, where 256 and 64 represent the input and output dimensions, respectively.

To assess the performance of different configurations, we employed tenfold cross-validation for each experiment. Figure 8 displays the average validation error obtained. By comparing the validation results of experiment Fig. 8(a) with different neuron numbers: 256-96-64, 256-128-64, 256-192-64, and 256-224-64, it is evident that the final evaluation error decreases as the number of neurons in the hidden layer increases. Specifically, configurations 256-192-64 and 256-224-64 demonstrate comparable performance but come at the expense of increased computational complexity.

In experiment Fig. 8(b), we explored the impact of increasing the number of hidden layers in the autoencoder while varying the number of neurons. Interestingly, adding hidden layers did not lead to a significant reduction in evaluation error, while it potentially increased computational complexity. It is worth noting that a neural network with at least one hidden layer has the ability to approximate any continuous function, ensuring that our autoencoder can accurately approximate the mapping relationship with arbitrary precision.



(a)

(b)

Fig. 8 Relationship between the training epoch and the evaluation error with various hidden layer and neuron counts

5 Conclusion and future work

In conclusion, the true value of an Image Secret Sharing (ISS) system lies in its ability to provide real-time security. In this study, we introduced a novel approach to randomized image secret sharing utilizing GPU-based parallel processing. By leveraging data-parallel tasks in both the sender and receiver pipelines, we achieved significant speedup improvements. Particularly, when dealing with larger image sizes and a higher number of shares, our GPU-based Randomized Image Secret Sharing (GRISS) method demonstrated much lower execution time increase compared to the traditional Randomized Image Secret Sharing (RISS) method, resulting in a speedup improvement of over twofold.

Additionally, the quality of the reconstructed image is crucial for any ISS. To address this, we employed a cutting-edge autoencoder specifically designed for single image super resolution. Through our experiments, we successfully demonstrated that the proposed GRISS method reconstructs the secured image while maintaining the original secret image's quality, further enhancing the value of the ISS.

In addition to the achievements and improvements presented in this research, there are several promising avenues for future exploration and development in the field of randomized image secret sharing. One potential direction for future research is to investigate the integration of machine learning algorithms into the image secret sharing process. Furthermore, exploring the potential of distributed computing and decentralized systems could lead to significant advancements in the scalability and robustness of image secret sharing protocols.

Author Contributions First author has designed the autoencoder part to incorporate super-resolution, second author has optimized GPU implementation, and the third author has supervised and reviewed the work.

Funding Open access funding provided by Manipal Academy of Higher Education, Manipal.

Availability of data and material (data transparency) Available.

Code availability (software application or custom code) Available.

Declarations

Conflict of interest/Competing interests No conflict of interest.

References

1. Blundo C, De Santis A, Naor M (2000) Visual cryptography for grey level images. Inf Process Lett 75(6):255–259
2. MacPherson L (2002) Grey level visual cryptography for general access structures. Master's thesis, University of Waterloo

3. Zhou Z, Arce GR, Di Crescenzo G (2006) Halftone visual cryptography. IEEE Trans Image Process 15(8):2441–2453
4. Hou Y-C (2003) Visual cryptography for color images. Pattern Recogn 36(7):1619–1629
5. Mhala NC, Jamal R, Pais AR (2017) Randomised visual secret sharing scheme for grey-scale and colour images. IET Image Process 12(3):422–431
6. Hou Y-C, Quan Z-Y, Tsai C-F, Tseng A-Y (2013) Block-based progressive visual secret sharing. Inf Sci 233:290–304
7. Kim C, Shin D, Leng L, Yang C-N (2018) Separable reversible data hiding in encrypted halftone image. Displays 55:71–79
8. Yan X, Lu Y (2019) Generalized general access structure in secret image sharing. J Vis Commun Image Represent 58:89–101
9. Kabirirad S, Eslami Z (2019) Improvement of (n, n)-multi-secret image sharing schemes based on boolean operations. J Inf Secur Appl 47:16–27
10. Yan B, Xiang Y, Hua G (2020) Improving visual quality for probabilistic and random grid schemes. In: Improving image quality in visual cryptography. Springer, pp 75–95
11. CUDA Zone | NVIDIA Developer (2020) https://developer.nvidia.com/cuda-zone. Accessed 10 June 2020
12. El-Shafai W, Ali AM, El-Nabi SA, El-Rabaie E-SM, Abd El-Samie FE (2023) Single image super-resolution approaches in medical images based-deep learning: a survey. Multimedia Tools and Applications, pp 1–37
13. Fukami K, Fukagata K, Taira K (2023) Super-resolution analysis via machine learning: a survey for fluid flows. Theoretical and Computational Fluid Dynamics, pp 1–24
14. Pandey G, Ghanekar U (2022) A conspectus of deep learning techniques for single-image super-resolution. Pattern Recog Image Anal 32(1):11–32
15. Kafri O, Keren E (1987) Encryption of pictures and shapes by random grids. Optics Lett 12(6):377–379
16. Shyu SJ (2009) Image encryption by multiple random grids. Pattern Recogn 42(7):1582–1596
17. Yan X, Liu X, Yang C-N (2018) An enhanced threshold visual secret sharing based on random grids. J Real-Time Image Process 14(1):61–73
18. Huang B-Y, Juan JS-T (2020) Flexible meaningful visual multi-secret sharing scheme by random grids. Multimedia Tools and Applications, pp 1–25
19. Naor M, Shamir A (1994) Visual cryptography. In: Workshop on the theory and application of cryptographic techniques. Springer, pp 1–12
20. Hou Y-C, Quan Z-Y (2011) Progressive visual cryptography with unexpanded shares. IEEE Transactions on Circuits and Systems for Video Technology 21(11):1760–1764
21. Mhala NC, Pais AR (2019) Contrast enhancement of progressive visual secret sharing (pvss) scheme for gray-scale and color images using super-resolution. Signal Process 162:253–267
22. Kang H, Leng L, Kim B-G (2022) Data hiding of multicompressed images based on shamir threshold sharing. Appl Sci 12(19):9629
23. Mhala NC, Pais AR (2021) A secure visual secret sharing (vss) scheme with cnn-based image enhancement for underwater images. The Visual Computer 37:2097–2111
24. SIPI Image Database (1999) https://sipi.usc.edu/database/. Accessed 10 June 2020
25. Index of / ychou/BPVSS (2012) mail.im.tku.edu.tw/ ychou/BPVSS/. Accessed 10 June 2020
26. Athar S, Wang Z (2019) A comprehensive performance evaluation of image quality assessment algorithms. IEEE Access 7:140030–140070

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Raviraja Holla M is working as Assistant Professor in the Department of Information and Communication Technology, Manipal Institute of Technology (a constituent institution of Manipal Academy of Higher Education), Manipal. He completed B.E.(CSE) from Bangalore University, India and M.Tech.(IT) from KSOU Mysore, India. He pursued his Ph.D. from the Department of Computer Engineering, National Institute of Technology Karnataka (NITK), Surathkal. His areas of interest include Information Security, High-Performance Computing, and Semantic Web.

Suma D is working as Assistant Professor in the Department of Computer Science and Engineering, Manipal Institute of Technology (a constituent institution of Manipal Academy of Higher Education), Manipal. She completed B.E.(ECE) from Kuvempu University, India and M.Tech.(CSE) from Visvesvaraya Technological University, India. Her areas of interest include Object Oriented Programming, High-Performance Computing, and Data Mining.

Alwyn Roshan Pais is a Professor in Department of Computer Science and Engineering, NITK Surathkal, India. He completed his B.Tech. (CSE) from Mangalore University, India, M.Tech. (CSE) from IIT Bombay, India and Ph.D. from NITK, India. His area of interest include Information Security, Image Processing and Computer Vision.