



A GPU scheme for multi-secret visual sharing with varied secret dimensions and contrast enhancement using blind super-resolution

M. Raviraja Holla¹ · D. Suma²

Received: 9 July 2023 / Accepted: 9 December 2023 / Published online: 19 January 2024
© The Author(s) 2024

Abstract Multi-secret visual sharing schemes are essential for secure and efficient sharing of sensitive visual information. However, existing schemes often overlook important considerations such as varied secret dimensions, contrast enhancement, and computational efficiency. This research addresses these challenges by proposing a comprehensive approach that addresses these limitations. Firstly, the scheme takes into account the varied dimensions of secret images encountered in real-world scenarios, allowing flexibility in sharing and reconstructing images of different sizes and aspect ratios. Secondly, the research integrates contrast enhancement techniques, such as blind super-resolution, to improve the visual quality and visibility of shared secret images affected by factors like noise, compression, or low lighting conditions. Lastly, to enhance the computational efficiency, the scheme leverages the power of Graphics Processing Units (GPUs) for parallel computing, enabling faster processing of large-scale image operations. The proposed scheme achieves outstanding results, with a high PSNR of 98.264, a strong NCC value of 0.965, an exceptionally low NAE value of 0.046, and an impressive SSIM value of 0.983. Furthermore, the GPU implementation provides

a remarkable overall speedup of 112× for 256 × 256 color images.

Keywords Multi-secret · Contrast · GPU · Security · Real-time

1 Introduction

Multi-secret visual sharing (MSVS) scheme is a cryptographic technique that allows multiple secret images to be divided into shares and distributed among different participants [20, 34, 44]. It is an extension of the visual secret sharing scheme, which is a method of splitting a secret image into shares that individually reveal no information about the original image [2, 7, 33]. In a multi-secret visual sharing scheme, two or more secret images are divided into shares in such a way that the shares for each secret image can be separately distributed [48]. The participants holding the shares can then collaboratively reconstruct the secret images by combining their shares [35]. The key advantage of this scheme is that it allows for the secure distribution and reconstruction of multiple secret images simultaneously. The process of creating and sharing the multi-secret shares involves several steps [25].

MSVS schemes can be classified according to various aspects and variations [18]. The three widely acknowledged classifications include Threshold-based schemes, Combinatorial schemes, and Adaptive schemes. The threshold-based schemes require a fixed threshold number of shares for each secret image, combinatorial schemes allow shares to contribute to the reconstruction of multiple secret images, and adaptive schemes dynamically adjust the reconstruction threshold based on conditions or requirements [4, 38, 39]. Each category has its own characteristics and offers

✉ M. Raviraja Holla
raviraj.holla@manipal.edu

D. Suma
suma.d@manipal.edu

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

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

different advantages depending on the specific needs of the application. MSVS has various applications, such as secure image transmission, secure storage, and secure multimedia communication [5]. It can be used in scenarios where multiple secret images need to be protected and securely shared among a group of authorized participants.

However, existing schemes often have limitations in terms of secret image dimensions and contrast levels [3, 16]. Additionally, enhancing the quality of shared secret images during the reconstruction process can further improve the user experience [22, 26]. To address these challenges, there is a need for a novel scheme that supports varied secret dimensions, incorporates contrast enhancement, and leverages the computational power of GPUs for efficient processing [47].

2 Literature survey

Multi-secret visual sharing (MSVS) is a method enabling the concealment of multiple confidential images within a collection of shared images. These shared images lack visual significance and can be disseminated among various participants. The retrieval of the confidential images is only achievable by consolidating a requisite number of share images. MSVS finds diverse applications, including image authentication, safeguarding, and watermarking.

The inception of MSVS can be traced back to 2005 when Feng et al. introduced an efficient secret sharing scheme based on Lagrange's interpolation for generalized access structures [19]. The complexity of the mathematical operations involved in the interpolation techniques during image reconstruction can be a significant drawback, especially when there is a high number of participants or when multiple secrets are shared simultaneously, exacerbating the computational burden. Chen and Wu (2011) [15] introduced an efficient $(n + 1, n + 1)$ MSVS utilizing Boolean-based sharing. This scheme ensures secret image confidentiality, enhances the capacity for sharing multiple secrets, reduces the demand for image transmission bandwidth, eases management overhead, and exhibits minimal computational cost and distortion in reconstructed secret images, as supported by experimental results. As a result, the scheme enhances the hiding capacity in image hiding schemes. A limitation of this scheme is the increased storage and transmission requirements due to the larger number of meaningless share images, potentially posing challenges in scenarios with limited storage space or bandwidth. Chen and Wu (2014) [13] presented a secure Boolean-based secret image sharing scheme that utilizes a random image generating function to increase the sharing capacity efficiently. The proposed scheme requires minimal CPU computation time for sharing or recovering secret images, with the bit shift subfunction being computationally

more demanding than the XOR subfunction. The additional computation required for generating the random image using the bit shift subfunction may result in higher processing time and resource utilization.

The invention by Chen et al. (2016) [14] unveils a novel, efficient, and secure secret image sharing scheme based on Boolean operations, showcasing its effectiveness through experimental results. One potential drawback of this scheme is that the computation time of the symmetric sharing-recovery function (SSRF) increases proportionally with the number of secret images, which may result in longer processing times for larger numbers of secrets. Existing literature on Boolean-based MSVS schemes is limited to $(2, n)$ and (n, n) cases, posing challenges for fault tolerance and practical applications. Kabirirad and Eslami [27] addressed this limitation by proposing a Boolean-based (t, n) -MSVS scheme for $t \geq 2, n$, and provides formal security proofs along with a comparison to existing literature. The drawback of this approach lies in its fault tolerance limitation, as the recovery of shared images becomes impossible if there are fewer than t participants, potentially compromising the scheme's robustness in specific scenarios.

Fulin et al. [30] presented a verifiable multi-secret sharing scheme that utilizes a symmetric binary polynomial based on the short integer solution problem. The scheme ensures verifiability without requiring interaction during distribution, reduces memory requirements, minimizes the size of shares per secret, and offers improved protection against quantum attacks. A drawback of this scheme is its vulnerability to quantum attacks, which could compromise the security and confidentiality of the shared secrets. In a recent study by Rawat et al. (2023) [40], an efficient MSVS was examined for diverse dimension images. The approach utilizes the Chinese Remainder Theorem (CRT), shift operations, and Exclusive-OR (XOR) operations to generate shares with enhanced randomness and resilience against attacks.

In their recent work, Wu et al. (2023) [46] introduced an innovative visually secure MSVS scheme. This scheme, which utilizes a combination of compressed sensing and secret image sharing, effectively tackles the limitations observed in current encryption methods. The proposed approach not only provides high compression and security but also ensures effective risk dispersion, contributing to the evolving landscape of secure image transmission techniques.

Sarkar et al. (2023) [41] have pioneered a novel technique focused on the secure distribution of k secret images among n participants, enabling collaborative recovery by k or more participants. Through the strategic utilization of spanning trees derived from a complete graph, represented by Prufer sequences, and the adoption of blockchain technology via the interplanetary file system for storing share images, their approach stands out for guaranteeing lossless reconstruction. This methodology has been substantiated

through extensive experimental analysis, confirming its robust security and cheat-proof attributes.

Recent progress in MSVS has yielded advancements in share expansion rates, bolstered security measures, and the capability to share diverse types of secrets [9, 42]. Share expansion rate denotes the proportion of share image size to secret image size, with lower rates being favorable for compact and efficient storage and transmission [28]. Improved security in MSVS schemes pertains to their resilience against various attacks, including collusion attacks and noise attacks. Furthermore, the enhanced capacity to share multiple secret types signifies the versatility of MSVS schemes, extending beyond images to encompass text, audio, and other forms of confidential information [31].

Our proposed research, informed by recent advancements in Multi-Secret Visual Cryptography Schemes (MSVS) outlined in the literature overview (refer to Table 1), strategically addresses key challenges in existing schemes. These challenges include handling varied secret dimensions, enhancing contrast for improved visibility, and leveraging GPU acceleration for efficient processing. Secret images in real-world scenarios often vary in dimensions, resolutions, and aspect ratios. To accommodate this diversity, our research aims to develop a scheme capable

of flexible sharing and reconstruction for varied secret dimensions. The research aims to significantly enhance the visual quality of shared secret images by integrating contrast enhancement techniques, such as blind super-resolution. Additionally, to address computational demands for large or simultaneous secret images, the emphasis is on GPU acceleration, harnessing parallel computing power for superior efficiency and speed in real-time or near-real-time scenarios.

3 Proposed model

Algorithm 1 comprehensively details the forward and backward stages of the proposed GPU method for MSVS, incorporating enhancements to the recovered image through the utilization of an autoencoder and Generative Adversarial Networks (GAN) blind super-resolution. The discussion on this algorithm will continue as follows. The forward phase of creating shares from secret images and the backward phase of reconstructing the secret images of various sizes are both explained in Sect. 3.1. We go over the suggested autoencoder and GAN in subsections 3.2.1 and 3.2.2 of Sect. 3.2 of our explanation of blind super-resolution.

Table 1 Literature overview

Author & Year	Methodology	Contributions	Limitations
Feng et al. (2005) [19]	Lagrange's interpolation	Efficient secret sharing	High computational burden
Chen and Wu (2011) [15]	Boolean-based sharing	Confidentiality, multiple secrets	Increased storage, transmission
Chen and Wu (2014) [13]	Random image generation	Secure secret sharing	Computational demands
Chen et al. (2016) [14]	Boolean operations	Novel and efficient scheme	Increased computation for multiple secrets
Kabirirad and Eslami (2018) [27]	Boolean-based (t, n) -MSVS	Fault tolerance	Recovery impossible if t participants are fewer
Fulin et al. (2023) [30]	Symmetric binary polynomial	Verifiable sharing	Vulnerability to quantum attacks
Rawat et al. (2023) [40]	CRT, shift operations, XOR	Enhanced randomness, resilience	Limited experimental validation, Potential scalability issues
Wu et al. (2023) [46]	Compressed sensing, secret sharing	High compression, security	Effective risk dispersion
Sarkar et al. (2023) [41]	Spanning trees, blockchain	Secure distribution, lossless reconstruction	Limited scalability, Blockchain overhead

Algorithm 1 MSVS with blind super-resolution

```

Data: Original secret images of varying dimensions
Result: Reconstructed secret images
// Forward phase
1 for each secret image do
2   Equalize secret images;
3   Apply Chinese Remainder Algorithm (CRA) to get CRT matrix;
4   CRT matrix split operation and two-Digit split with left shift;
5   Using Key image, Generate Shares
// Backward phase
6 Key Image XOR Operation;
7 Shared Images XOR Operation to Recover Split Images ;
8 Generate CRT Matrix to get Equalized images ;
9 for each shared image do
10  Autoencoder;
11  Generative Adversarial Network (GAN);
12 end
    
```

3.1 Multi-varied image sharing

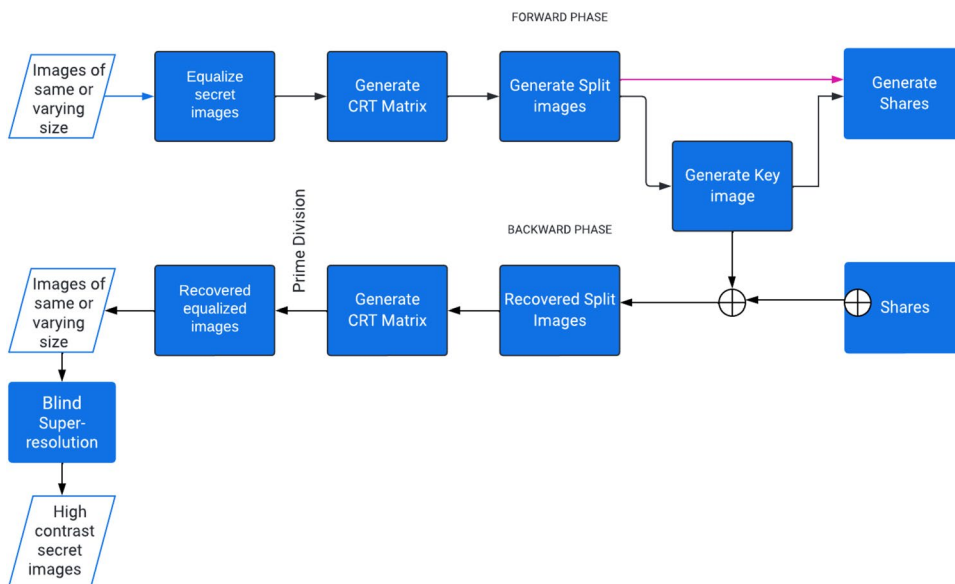
The proposed method aims to offer a new secret sharing mechanism for secret images of various dimensions. It utilizes the Chinese Remainder Theorem (CRT) for generating shares and reconstructing the secret. The process for share generation is illustrated in Fig. 1 containing forward and backward phases.

The secret images of varying dimensions are fed into the "Equalize secret images" block, which generates secret images of the same dimension, as outlined in Algorithm 3. Algorithm 2, the Chinese Remainder Algorithm (CRA), is applied to the equalized secret images, resulting in the

CRT matrix described in Algorithm 4. The process begins with the CRT matrix undergoing a two-digit split operation, followed by a one-bit left shift operation, resulting in split images as described in Algorithm 5. By applying Algorithm 6, the key image is obtained through the XOR operation performed on the split images. Subsequently, shared images of the same dimensions are generated by applying the XOR operation between the key image and the split images, following Algorithm 7.

To restore the original secrets, the reconstruction process is carried out on the generated shares, as illustrated in Fig. 1. To obtain the split images, the XOR operation is employed between the shared images and the key image.

Fig. 1 Visualization of the proposed GPU scheme



Afterwards, the split images are combined in a digit-wise manner, accompanied by a one-bit right shift operation, resulting in the formation of the CRT matrix. The complete secret images are then reconstructed by applying the prime division procedure to the CRT matrix. Ultimately, the secret images of varying dimensions are recovered. To conserve space, we have omitted algorithms on the reverse process, as it is essentially the inverse of the forward phase.

By employing GPU parallel algorithms in both the forward and backward stages, we harnessed the computational power required for these steps. This approach aimed at fostering sustainable development growth through the efficient utilization of data-parallel tasks.

3.2 Blind Super-Resolution (BSR)

Algorithm 2 Chinese remainder algorithm (CRA)

Input: n congruences of the form $a[i] \equiv x \pmod{m[i]}$ for $i = 1$ to n
Output: Solution to the CRT in the form $x \equiv result[i] \pmod{m[i]}$ for $i = 1$ to n

- 1 Read the number of congruences (n) and the congruences in the form $a[i] \equiv x \pmod{m[i]}$ for $i = 1$ to n .
- 2 Allocate device memory for d_a , d_m , and d_result . Declare device memory pointers d_a , d_m , and d_result .
- 3 Copy the input congruences $a[i]$ and moduli $m[i]$ from host to device memory.
- 4 Calculate the grid and block dimensions for parallel execution
// Kernel function.
- 5 Define the kernel function `chineseRemainderTheorem(d_a, d_m, n, d_result)` to perform the parallel CRT calculation:
 - (i) Each thread calculates its unique index $tid = blockIdx.x * blockDim.x + threadIdx.x$
 - (ii) **if** $tid < n$: **then**
 - (a) Calculate the product of all moduli except the current one.
 - (b) Find the modular multiplicative inverse of the product modulo $m[tid]$
 - (c) Calculate the CRT solution using the given formula:
 $result[tid] = (a[tid] * product * inversed) \% (m[0] * m[1] * m[2] * \dots * m[n-1])$
- 6 Launch the kernel function `chineseRemainderTheorem` with the configured grid and block dimensions.
- 7 Copy the result from device to host memory
- 8 Exit

Algorithm 3 Generate secrets of equal dimension

Input: n input secrets (I), Number of secrets (n) and the dimensions ($r1, r2, \dots, rn$) and ($c1, c2, \dots, cn$) for each secret.
Output: n secrets (E) of equal dimension

- 1 Determine the maximum number of rows (N) and columns (M) among all the secret images.
- 2 Allocate memory for the input secrets $I1, I2, \dots, In$, and the output secrets $E1, E2, \dots, En$ on the GPU.
- 3 Copy the input secrets $I1, I2, \dots, In$ from the host (CPU) to the device (GPU) memory.
- 4 Launch a CUDA kernel with $(N \times M \times n)$ threads.
// kernel function
 - (i) $linearIdx = blockIdx.x * blockDim.x * blockDim.y + threadIdx.y * blockDim.x + threadIdx.x$.
 - (ii) **for each secret image** Ik **do**
 - if** $rIk > linearIdx / M$ or $cIk > linearIdx \% M$ **then**
 - | $E(linearIdx) = rand() \times 255$.
 - end**
 - end**
 - $E(linearIdx) = Ik(linearIdx)$
- 5 Exit

Algorithm 4 Generate CRT matrix

```

Input: n secret images (E) of equal size N x M.
Output: CRT matrix (CM) of size N x M
// Let p1, p2, ... pn be the coprime numbers > 255.
1 Allocate memory for the secret images (E), CRT matrix (CM), and coprime numbers
  (p) on the GPU.
2 Copy the secret images (E) and coprime numbers (p) from the host to the GPU
  memory.
3 Launch a CUDA kernel with (N x M x n) threads.
// Kernel function
(i) linearIdx =
    blockIdx.x x blockDim.x x blockDim.y + threadIdx.y x blockDim.x + threadIdx.x.
(ii) for each secret image Ek do
    | Compute CRA(E1[linearIdx], E2[linearIdx], ..., En[linearIdx], p1, p2, ..., pn) to
    | obtain CM[linearIdx].
    end
4 Exit
  
```

Algorithm 5 Generate split images

```

Input: CM matrix of size N x M.
Output: Split images p1, p2, ..., pt
// Let the maximum value of t can be t = (P x n)/2 where P denotes the
  digit size of the largest co-prime number and n denotes the number of
  secret images.
1 Define the dimensions of the CM matrix, N (number of rows) and M (number of
  columns), and the number of split images t.
2 Allocate memory for the CM matrix (input) and the split images P (output) on the
  GPU.
3 Copy the CM matrix from the host (CPU) to the device (GPU) memory.
4 Launch a CUDA kernel with t x N x M threads.
5 Compute the linear index of the thread within the grid: linearIdx = blockIdx.x *
  blockDim.x + threadIdx.x + blockIdx.y * blockDim.y + threadIdx.y.
6 if linear index is within the valid range (N x M) then
7   | Temp = CM[linearIdx].
7   | // 1 >= k <= t
8   | Pk[linearIdx] = CM[linearIdx] % 100 << 2
9   | Temp = CM[linearIdx] / 100
10 end
11 Exit
  
```

Fig. 2 Visualization of the proposed architectural design

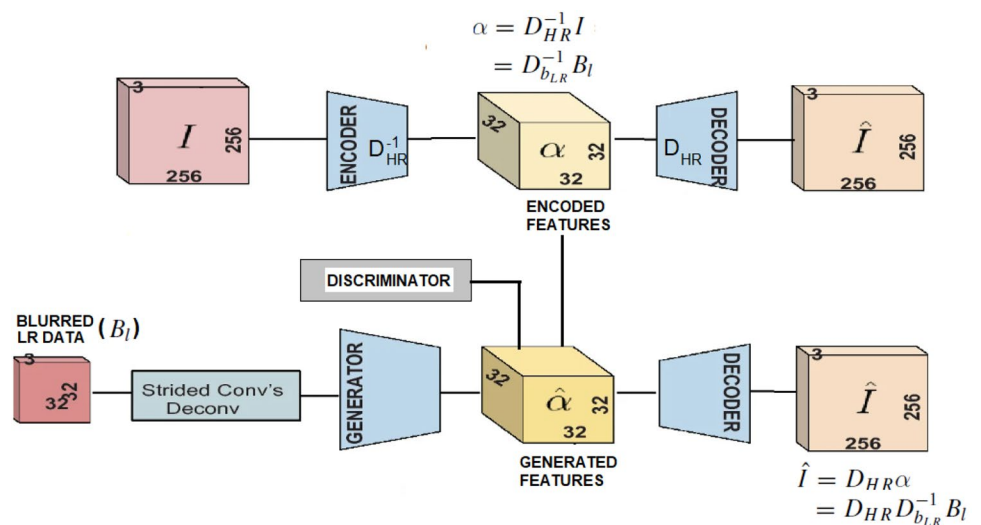
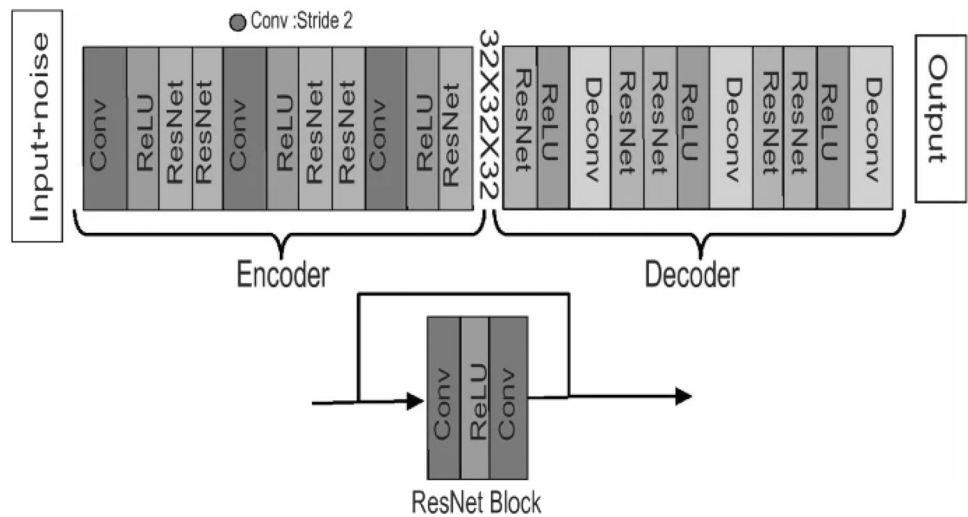


Fig. 3 Autoencoder architecture incorporating residual networks



Algorithm 6 Generate key image

```

Input: Split images p1, p2, ..., pt.
Output: Key image K with N × M dimension.
1 Define the dimensions of the split images, N (number of rows) and M (number of
  columns), and the number of split images t.
2 Allocate memory for the split images p1, p2, ..., pt, and the key image K on the
  GPU. Copy the split images p1, p2, ..., pt from the host (CPU) to the device
  (GPU) memory.
3 Launch a CUDA kernel with N × M threads.
4 linearIdx =
  blockIdx.x × blockDim.x × blockDim.y + threadIdx.y × blockDim.x + threadIdx.x.
  if linear index is within the valid range (N × M). then
5   | K[linearIdx] = p1[linearIdx] ⊕ p2[linearIdx] ..., pt[linearIdx].
6 end
7 Exit
    
```

Algorithm 7 Generate shares

```

Input: Split images p1, p2, ..., pt and key image K (N × M).
Output: Shares s1, s2, ..., st.
1 Define the dimensions of the split images, N (number of rows) and M (number of
  columns), and the number of split images t.
2 Allocate memory for the split images p1, p2, ..., pt, the key image K, and the shares
  s1, s2, ..., st on the GPU.
3 Copy the split images p1, p2, ..., pt and the key image K from the host (CPU) to
  the device (GPU) memory.
4 Launch a CUDA kernel with N × M threads.
5 linearIdx = blockIdx.x × blockDim.x × blockDim.y + threadIdx.y × blockDim.x +
  threadIdx.x.
6 if linear index is within the valid range (N × M) then
7   | if t is even then
8     | Si[linearIdx] = pi[linearIdx] ⊕ K[linearIdx].
9   | end
10  | if t is odd then
11    | S1[linearIdx] = p1[linearIdx] ⊕ R ⊕ K[linearIdx].
12    | St[linearIdx] = pt[linearIdx] ⊕ R ⊕ K[linearIdx].
13    | if linearIdx ≠ 1 or t then
14      | Si[linearIdx] = pi[linearIdx] ⊕ K[linearIdx].
15    | end
16  | end
17 end
18 Exit
    
```

The process of reconstructing a high-resolution (HR) image from degraded low-resolution (LR) images, without prior knowledge of the specific degradation phenomena such as blurring or additive noise, is referred to as blind super-resolution (BSR) reconstruction [36]. BSR serves as an appropriate technique to enhance the quality of the reconstructed image, particularly when the degradation phenomena in the proposed model are unfamiliar and challenging to estimate during the super-resolution reconstruction process.

The proposed approach involves utilizing autoencoders (AE) to learn a strong representation of HR image-domain information, which is crucial. Autoencoders have demonstrated remarkable success in unsupervised learning by effectively encoding data into a compact form. Once a robust representation is acquired for clean HR patches, the subsequent step involves generating an invariant representation from blurred LR data. To accomplish this, the proposal suggests employing a GAN framework, where a generator and discriminator engage in a competitive training process. The generator’s objective is to deceive the discriminator by producing clean features from blurred LR data that closely resemble those generated by the autoencoder, thereby achieving invariance. On the other hand, the discriminator aims to outperform the generator by distinguishing between clean and blurred features.

Fig. 2 displays the schematic of the proposed architecture. The primary architectural distinction lies in the generator, which is now tasked with conducting joint super-resolution (SR) and deblurring. Due to the disparity in dimensions between the LR input and the HR image, we introduce fractional strided convolutions during the initial phases of the generator to address this distinction.

3.2.1 Autoencoder

The Auto Encoder (AE) maps input data to a reduced-dimensional space and subsequently reconstructs it from this encoded representation [43]. Incorporating residual blocks (ResNet) in the AE architecture led to faster convergence and improved output quality [24]. Residual blocks bypass higher-level features to the output and address the vanishing gradient problem [45]. To ensure encoder reliability and prevent learning an identity map, we introduced noise corruption in the training data (30% of the time). The architecture, including the ResNet block, is depicted in Fig. 3. Equations (1) and (2) provide a comprehensive description of the filter and feature map sizes, as well as the stride values employed in both the encoder and decoder.

$$Encoder : A_{3 \rightarrow 8}^5 \downarrow 2 \rightarrow B_8^{5(2)} \rightarrow A_{8 \rightarrow 16}^5 \downarrow 2 \rightarrow B_{16}^{5(2)} \rightarrow A_{16 \rightarrow 32}^3 \downarrow 2 \rightarrow B_{32}^3 \tag{1}$$

$$Decoder : B_{32}^3 \rightarrow A_{32 \rightarrow 16}^2 \uparrow 2 \rightarrow B_{16}^{5(2)} \rightarrow A_{16 \rightarrow 8}^4 \uparrow 2 \rightarrow B_8^{5(2)} \rightarrow A_{8 \rightarrow 3}^4 \uparrow 2 \tag{2}$$

The provided notation uses the following representation: $A_{p \rightarrow q}^r \downarrow s$ indicates a down-convolution operation that maps a feature dimension from p to q using a stride of s and a filter size of r . Similarly, the symbol \uparrow denotes an up-convolution. On the other hand, $B_p^{q(r)}$ refers to the residual block, which consists of a convolutional layer followed by a ReLU activation block. The residual block has an output feature size of p , a filter size of q , and r indicates the count of iterations for the residual blocks.

3.2.2 Feature mapping using Generative Adversarial Networks (GAN)

GAN consists of two models: an entity called the Generator \mathcal{G} and another entity known as the Discriminator \mathcal{D} [11]. The Discriminator’s objective is to differentiate between samples generated by \mathcal{G} and the training data samples, while the Generator’s aim is to outsmart the Discriminator by producing samples that closely resemble the actual data distribution [29]. We used a conditional GAN. The training of conditional GANs is guided by the modified mini-max cost function, as expressed in Equation (3) [17].

$$\min_{\mathcal{G}} \max_{\mathcal{D}} C_{cond}(\mathcal{G}, \mathcal{D}) = E_{i,j \sim P_{data}(i,j)} [\log \mathcal{D}(i,j)] + E_{i \sim P_{data}(i), k \sim P_k(k)} [\log(1 - \mathcal{D}(i, \mathcal{G}(i,k)))] \tag{3}$$

The main goal is to generate a specific category of natural images based on the input random vector k [21]. In this context, $\mathcal{D}(i)$ represents the discriminator’s assigned probability to input i , while j denotes the clean target feature used for discerning whether i is a real sample. Additionally, k represents the input random vector. The probability distributions of the data i and the input random vector k are denoted as P_{data} and P_k , respectively. In conditional GANs, the generator’s objective is to capture the joint probability distribution of i and k to effectively represent the data distribution [23] [37]. For our specific task, training the network without considering k results in the network acquiring a mapping from i to a deterministic output j , which corresponds to the clean feature.

During the training process, we follow the conventional procedure for training the GAN. However, instead of instructing the discriminator to differentiate between generated images and clean images, we instruct it to distinguish between their corresponding features. The generator

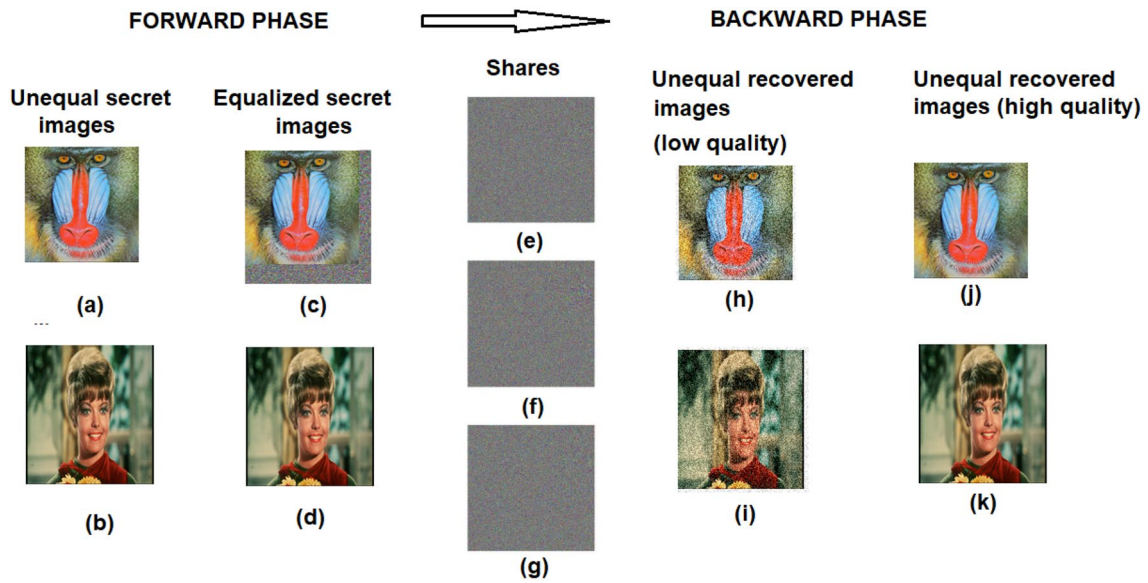


Fig. 4 Transitioning images from the forward phase to the backward phase

Fig. 5 Colour secret images of size (256 × 256) a Female, b Couple, c House, d Tree

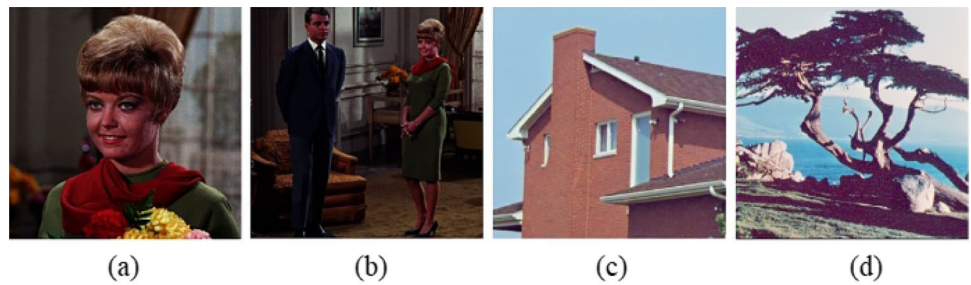
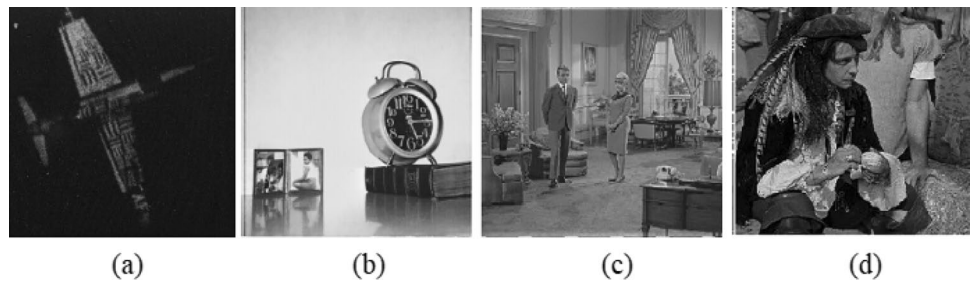


Fig. 6 Grayscale secret images of size (256 × 256) (a) Airplane, b Clock, c Couple, d Man



architecture (4×) and the discriminator architecture are provided in Equations (4) and (5).

During the training process, we employed the commonly favored reconstruction cost. The reconstruction cost,

$$\begin{aligned}
 A_{3 \rightarrow 8}^5 \uparrow 2 \rightarrow C_{8 \rightarrow 8}^5 \rightarrow A_{8 \rightarrow 16}^5 \uparrow 2 \rightarrow A_{16 \rightarrow 16}^5 \downarrow 2 \rightarrow B_{16}^{5(2)} \rightarrow A_{16 \rightarrow 32}^5 \downarrow 2 \rightarrow B_{32}^{5(2)} \\
 \rightarrow \hat{A}_{32 \rightarrow 32}^3 \downarrow 2 \rightarrow B_{32}^{5(2)} \rightarrow A_{32 \rightarrow 128}^3 \downarrow 2 \rightarrow B_{128}^{3(2)} \rightarrow \hat{A}_{128 \rightarrow 32}^3 \uparrow 2
 \end{aligned} \tag{4}$$

$$A_{32 \rightarrow 32}^5 \rightarrow A_{32 \rightarrow 32}^5 \downarrow 2 \rightarrow A_{32 \rightarrow 16}^5 \rightarrow A_{16 \rightarrow 16}^5 \downarrow 2 \rightarrow A_{16 \rightarrow 8}^5 \rightarrow A_{8 \rightarrow 8}^3 \downarrow 2 \rightarrow A_{8 \rightarrow 1}^3 \tag{5}$$

Table 2 Assessing the proposed scheme's objective quality compared to benchmarks

Images	PSNR (dB)			NCC			NAE			SSIM		
	[46]	[41]	Proposed	[46]	[41]	Proposed	[46]	[41]	Proposed	[46]	[41]	Proposed
Fig. 5(a)	6.203	8.731	28.201	0.801	0.816	0.881	0.616	0.263	0.092	0.615	0.712	0.891
Fig. 5(b)	6.856	8.627	29.945	0.756	0.773	0.931	0.263	0.241	0.089	0.724	0.810	0.918
Fig. 5(c)	7.541	7.983	29.462	0.741	0.822	0.953	0.431	0.224	0.019	0.052	0.787	0.983
Fig. 5(d)	6.828	8.287	27.549	0.771	0.782	0.872	0.384	0.268	0.035	0.787	0.746	0.921
Fig. 6(a)	6.623	8.623	29.571	0.626	0.889	0.935	0.298	0.209	0.046	0.754	0.762	0.907
Fig. 6(b)	6.062	7.872	29.502	0.594	0.697	0.894	0.255	0.243	0.093	0.794	0.695	0.910
Fig. 6(c)	7.717	8.890	28.295	0.730	0.751	0.928	0.243	0.222	0.084	0.867	0.761	0.927
Fig. 6(d)	6.820	7.131	28.106	0.707	0.839	0.965	0.280	0.371	0.104	0.783	0.836	0.895

quantified as the mean squared error (MSE) loss, calculates the l_2 distance between the expected and observed images. It serves as a measure of the dissimilarity between the generated output and the target output, aiding in minimizing the reconstruction error.

To train the autoencoder, we utilized a subset of images available at data repository [32], which encompasses approximately 202,599 images. These images were resized to dimensions of 256×256 . Randomly, we selected 200,000 samples for the training set, while the remaining images were allocated to the test and validation sets. To facilitate the learning of meaningful data representations, the inputs were subject to random corruption through the addition of Gaussian noise, with a standard deviation of 0.2, applied 30% of the time.

We employed the Adam optimizer [20] with an initial learning rate of 0.0002 and a momentum of 0.9. A batch size of 16 was utilized for training. The training process required approximately 3×10^5 iterations to converge. To prevent the final results from being overly sharpened, the gradient cost was scaled by a factor of $\lambda = 0.1$.

In the second stage of our training process, our primary objective was to acquire a representation that remains unaffected by changes in blur and resolution using blurred LR data. To accomplish this, we synthesized blurred face data by applying parametric blur kernels, which possess space-invariant properties. The blur intensity was controlled by adjusting length (l) and angle (θ). Within our experimentation, we explored a range of blur variations, specifically $l \in (0, 40)$ pixels and $\theta \in (0^\circ, 180^\circ)$. To construct the training sets for different super-resolution (SR) factors, we took the pristine images and applied the parametrized blur kernels, followed by downsampling with factors of 2, 4, and 8. Each set comprised a substantial number of samples, totaling 400,000 blurred LR training data instances. Moving on to the first stage of the generator, we employed learnable up-convolution filters to upscale the input data to a dimension of 256.

To enhance the stability of the GAN training, we employed a technique called smooth labeling, as discussed in [6], for both the blur and clean features. In the initial phase, the training focused solely on the feature costs for approximately 100,000 iterations, with a weight factor of $\lambda_{\text{adv}} = 0.001$ and $\lambda_1 = 1$. Subsequently, fine-tuning of the generator was carried out by introducing the mean squared error (MSE) cost and reducing the weight assigned to the adversarial cost. Specifically, the weight factors used were $\lambda_2 = 1$, $\lambda_1 = 1$, and $\lambda_{\text{adv}} = 0.0001$.

4 Experimental result analysis

The CUDA, OpenCV, and MATLAB were employed to develop and evaluate the proposed GPU model on the PARAM Shavak Super Computer. The PARAM Shavak is equipped with an Intel®Xeon®-E5-2670 CPU, which features two physical processors, each consisting of 24 cores with a clock speed of 2.3 GHz. The system is further equipped with a massive 8TB RAM configuration to support computational requirements. Additionally, a single NVIDIA®Tesla K40 GPU with 2880 cores, 12 GB GDDR5, and a clock speed of 745 MHz was utilized.

We employed the USC-SIPI image database [1] in our research investigations. The transformation process of input images with different sizes (Figs. 4 (a) and (b)) is visually depicted in Fig. 4. The figure demonstrates the steps of the forward and backward phases, with the progression from left to right.

For experimental purposes, Figs. 5 and 6 display sample images that have been rescaled from their original 256×256 dimensions. These rescaled images vary in size, serving the purpose of the experiment. Section 4.1 assesses the performance of the proposed scheme using objective parameters. Furthermore, in Sect. 4.2, the advantages of the proposed scheme are discussed, highlighting a few key factors. The analysis of the speedup achieved by the proposed scheme is presented in Sect. 4.3.

Table 3 Advantages of the proposed scheme compared to existing schemes

Factors	[15]	[13]	[14]	[12]	[10]	Proposed
Sharing type	(n+1,n+1)	(n,n)	(n,n)	(n,n)	(ti,ti)	(n,t)
Image type	Grayscale	Grayscale	Grayscale	Grayscale	Colored	Grayscale/Colored
Sharing capability	n/(n+1)	n/n	n/n	n/n	n/n+1	n/t
Dimension type	Fixed	Fixed	Fixed	Varying	Fixed	Varying
Pixel scaling	No	No	No	Yes	No	Yes
Restoring approach	XOR	XOR	XOR	XOR	XOR & MA	CRA
GPU-based	No	No	No	No	No	Yes
Super-resolution utilization	No	No	No	No	No	Yes

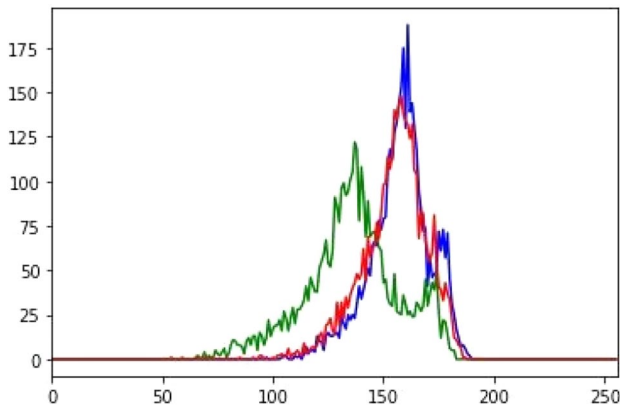


Fig. 7 Rescaled input secret image RGB histogram analysis (Size: 64 × 64)

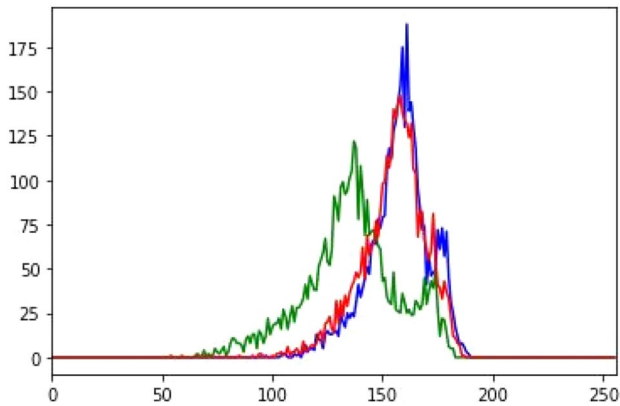


Fig. 8 Rescaled recovered secret image RGB histogram analysis (Size: 64 × 64)

4.1 Objective evaluation of the proposed model’s performance

We employed the following objective performance analysis parameters to demonstrate the superior performance of the proposed model compared to two advanced models, namely

Table 4 GPU scheme’s forward phase speed enhancements

Sl.No.	Forward phase activities	Execution time in seconds		Speedup
		CPU model	Proposed GPU model	
1.	Generate secrets of equal dimension.	1.327010	0.002984	445
2.	Generate CRT matrix.	2.893001	0.053811	54
3.	Generate split images.	1.211613	0.002818	430
4.	Generate key image.	1.238713	0.007561	164
5.	Generate shares.	1.247594	0.003671	340

Note: For 256 × 256 colour images and four shares

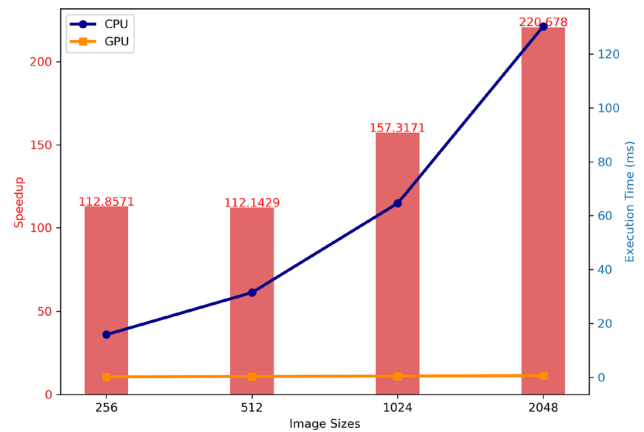


Fig. 9 Comparing speedup and execution time across various image sizes with four shares

Wu et al. (2023) [46] and Sarkar et al. (2023) [41]. Let W and H denote the width and height of the image respectively. Let Equations (6) and (7) represent the input and recovered images, where $i(m, n)$ and $r(m, n)$ correspond to the intensities of the pixels at (m, n) in I and R respectively.

$$I = \{i(m, n) | (0 \leq m \leq W - 1), (0 \leq n \leq H - 1)\} \tag{6}$$

$$R = \{r(m, n) | (0 \leq m \leq W - 1), (0 \leq n \leq H - 1)\} \tag{7}$$

- **Peak Signal to Noise Ratio (PSNR):** The Mean Square Error (MSE), defined in Equation (8), serves as a measure of effectiveness. A lower MSE indicates a higher level of similarity between the reference and target images, signifying improved effectiveness. The Root Mean Squared Error (RMSE) and PSNR, defined in Equations (9) and (10) respectively, further contribute to assessing image quality. A higher PSNR value signifies superior image quality, emphasizing the positive alignment between the two. The proposed scheme reached a maximum PSNR of 29.945dB, as demonstrated in Table 2.

$$MSE(I, R) = \frac{1}{WH} \sum_{m=0}^{W-1} \sum_{n=0}^{H-1} [i(m, n) - r(m, n)]^2 \tag{8}$$

$$RMSE(I, R) = \sqrt{MSE(I, R)} \tag{9}$$

$$PSNR(I, R) = 20 \cdot \log_{10} \left(\frac{255}{RMSE(I, R)} \right) \tag{10}$$

- **Normalized Cross-Correlation (NCC):** The NCC, defined in Equation (11), provides a measure of similarity between two images. A higher value of NCC indicates a greater degree of similarity between the images, thus serving as an indicator of their likeness. Table 2 reveals a NCC value of 0.965, showcasing the performance of the proposed scheme.

$$NCC(I, R) = \sum_{m=0}^{W-1} \sum_{n=0}^{H-1} \left[\frac{I(m, n) \times R(m, n)}{I^2(m, n)} \right] \tag{11}$$

- **Normalized Absolute Error (NAE)** The NAE, as defined in Equation (12), quantifies the error between the secret and recovered images. A lower NAE value signifies a higher quality of the recovered image. Therefore, minimizing the NAE serves as a measure of the effectiveness of the recovery process in achieving better image quality.

$$NAE(I, R) = \sum_{m=0}^{W-1} \sum_{n=0}^{H-1} \left[\frac{I(m, n) - R(m, n)}{I(m, n)} \right] \tag{12}$$

The proposed scheme exhibits an NAE value of 0.046, as evidenced by the data presented in Table 2.

- **Structural Similarity Index (SSIM):** The SSIM is a spatial measure that compares the luminance, contrast, and structure of two images originating from the same source [8]. It provides a numerical value between -1 and 1, where an SSIM value of 1 indicates a perfect match

between the compared images. Equation (13) defines the calculation for SSIM, taking into account various components to assess the similarity between the images. As indicated in Table 2, the proposed scheme achieves an SSIM value of 0.983, highlighting its effectiveness.

$$SSIM(I, R) = [l(I, R)]^\alpha \cdot [c(I, R)]^\beta \cdot [s(I, R)]^\gamma \tag{13}$$

Where,

$$l(I, R) = \frac{2\mu_I\mu_R + C_1}{\mu_I^2 + \mu_R^2 + C_2}$$

$$c(I, R) = \frac{2\sigma_I\sigma_R + C_2}{\sigma_I^2 + \sigma_R^2 + C_2}$$

$$s(I, R) = \frac{\sigma_{IR} + C_3}{\sigma_I\sigma_R + C_3}$$

Equation (14) is the reduced form of Equation (13), assuming $\alpha = \beta = \gamma = 1$ and $C3 = C2/2$.

$$SSIM(I, R) = \frac{(2\mu_I\mu_R + C_1)(2\sigma_{IR} + C_2)}{(\mu_I^2 + \mu_R^2 + C_1)(\sigma_I^2 + \sigma_R^2 + C_2)} \tag{14}$$

Here, σ_I and σ_R represent the local standard deviations of the input image I and the recovered image R , respectively. Similarly, μ_I and μ_R denote the means of I and R , while σ_{IR} represents the cross-covariance between the two images.

- **RGB histogram analysis:** Figures 7 and 8 demonstrate the resemblance between the RGB histograms of the rescaled recovered image and the original image, highlighting the superior quality of the proposed model.

4.2 Advantages of the proposed scheme compared to existing schemes

This section highlights the notable advantages of the proposed scheme over the existing schemes outlined in Table 3. The proposed (n, t) scheme is characterized by its flexibility, as 'n' represents the number of secret images and 't' can be equal to, greater than, or less than 'n'. This proposed approach is applicable to both grayscale and colored images. The sharing capability is determined by dividing the number of secret images by the number of shared images. Hence, the sharing capacity of the proposed scheme is represented by 'n/t'. Notably, the proposed approach effectively handles secret images with different dimensions, requiring pixel scaling when the sizes differ. To recover the secret images, the computationally expensive CRA method is employed. Furthermore, the proposed scheme utilizes the computational capabilities of GPUs, enabling its suitability for real-time applications. The inclusion of blind super-resolution enhances the quality of the recovered image, approximating it to the original secret image.

4.3 Speedup of the proposed scheme

Table 4 presents a comparison of execution times for the forward phase steps using the CPU (sequential) and GPU (parallel) versions of the proposed scheme, focusing on 256×256 color images. The speedup is calculated as the ratio of CPU execution time to GPU execution time. The total execution time for the CPU model is 7.91793 s, while the GPU version takes only 0.070845 s, resulting in an impressive overall speedup of 112×. The backward phase maintains similar execution times, thus preserving the speedup.

Fig. 9 depicts the progressive increase in speedup with larger color image sizes, employing four shares. The slight rise in GPU execution time demonstrates the scalability of the GPU version as the image size grows.

5 Conclusion and future work

In conclusion, this research has successfully addressed key challenges in multi-secret visual sharing schemes, focusing on varied secret dimensions, contrast enhancement, and GPU acceleration. By developing a scheme capable of handling secret images with diverse dimensions, flexibility in sharing and reconstructing images of different sizes and aspect ratios has been achieved. The integration of contrast enhancement techniques, such as blind super-resolution, has significantly improved the visual quality and visibility of shared secret images, enhancing their interpretability and usability. Furthermore, the utilization of GPU acceleration techniques has boosted the computational efficiency and speed of the multi-secret image sharing scheme, making it suitable for real-time or near-real-time applications.

Future work could explore dynamic adaptation to image characteristics and extension to three-dimensional images. Additionally, optimizing the scheme for real-time processing, enhancing security, and assessing scalability for large-scale deployment are potential avenues for further research.

Author contributions The work is carried out by the first author under the guidance of the second author.

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

Availability of data and materials Available.

Code availability Available

Declarations

Conflict of interest No conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. SIPI Image Database (1999(accessed August 23, 2020)). sipi.usc.edu/database/database.php?volume=misc?
2. Agarwal A, Deshmukh M (2021) 3-d plane based extended shamir's secret sharing. *Int J Inf Technol* 13:609–612
3. Ahuja B, Doriya R (2022) Bifold-crypto-chaotic steganography for visual data security. *Int J Inf Technol* 14(2):637–648
4. Alkhodaidi TM, Gutub AA (2022) Scalable shares generation to increase participants of counting-based secret sharing technique. *Int J Inf Comput Secur* 17(1–2):119–146
5. Alshathri S, Hemdan EED (2023) An efficient audio watermarking scheme with scrambled medical images for secure medical internet of things systems. *Multim Tools Appl* 82(13):20177–95
6. Arjovsky M, Bottou L (2017) Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*
7. Arora A, Garg H, Shivani S, et al (2023) Privacy protection of digital images using watermarking and qr code-based visual cryptography. *Adv Multim* 2023
8. Athar S, Wang Z (2019) A comprehensive performance evaluation of image quality assessment algorithms. *Ieee Access* 7:140030–140070
9. Bachiphale PM, Zulpe NS (2023) Optimal multiset image sharing using lightweight visual sign-cryptography scheme with optimal key generation for gray/color images. *International Journal of Image and Graphics* p. 2550017
10. Bisht K, Deshmukh M (2021) A novel approach for multilevel multi-secret image sharing scheme. *J Supercomput* 77(10):12157–12191
11. Brophy E, Wang Z, She Q, Ward T (2023) Generative adversarial networks in time series: A systematic literature review. *ACM Comput Surv* 55(10):1–31
12. Chen CC, Chen JL (2017) A new boolean-based multiple secret image sharing scheme to share different sized secret images. *J Inform Sec Appl* 33:45–54
13. Chen CC, Wu WJ (2014) A secure boolean-based multi-secret image sharing scheme. *J Syst Softw* 92:107–114
14. Chen CC, Wu WJ, Chen JL (2016) Highly efficient and secure multi-secret image sharing scheme. *Multim Tools Appl* 75:7113–7128
15. Chen TH, Wu CS (2011) Efficient multi-secret image sharing based on boolean operations. *Signal Process* 91(1):90–97
16. Cheng J, Yan X, Liu L, Jiang Y, Wang X (2022) Meaningful secret image sharing with saliency detection. *Entropy* 24(3):340
17. Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA (2018) Generative adversarial networks: An overview. *IEEE Signal Process Mag* 35(1):53–65

18. De Prisco R, De Santis A, Palmieri F (2023) Bounds and protocols for graph-based distributed secret sharing. *IEEE Transactions on Dependable and Secure Computing*
19. Feng JB, Wu HC, Tsai CS, Chu YP (2005) A new multi-secret images sharing scheme using Lagrange's interpolation. *J Syst Softw* 76(3):327–339
20. Francis N, Monoth T (2023) Security enhanced random grid visual cryptography scheme using master share and embedding method. *Int J Inf Technol* 15(7):3949–3955
21. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. *Commun ACM* 63(11):139–144
22. Gutte VS, Parasar D (2022) Sailfish invasive weed optimization algorithm for multiple image sharing in cloud computing. *Int J Intell Syst* 37(7):4190–4213
23. Han J, Wang D, Li Z, Dey N, Crespo RG, Shi F (2023) Plantar pressure image classification employing residual-network model-based conditional generative adversarial networks: a comparison of normal, planus, and talipes equinovarus feet. *Soft Comput* 27(3):1763–1782
24. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778
25. Huang BY, Juan JST (2020) Flexible meaningful visual multi-secret sharing scheme by random grids. *Multim Tools Appl* 79(11–12):7705–7729
26. Jisha T, Monoth T (2023) Contrast-enhanced visual cryptography schemes based on block pixel patterns. *Int J Inform Technol*. <https://doi.org/10.1109/CW.2010.39>
27. Kabirirad S, Eslami Z (2018) A (t, n)-multi secret image sharing scheme based on boolean operations. *J Vis Commun Image Represent* 57:39–47
28. Kang Y, Kanwal S, Pu S, Liu B, Zhang D (2023) Ghost imaging-based optical multilevel authentication scheme using visual cryptography. *Opt Commun* 526:128896
29. Kar MK, Neog DR, Nath MK (2023) Retinal vessel segmentation using multi-scale residual convolutional neural network (msr-net) combined with generative adversarial networks. *Circuits Syst Signal Process* 42(2):1206–1235
30. Li F, Yan J, Zhu S, Hu H (2023) A verifiable multi-secret sharing scheme based on short integer solution. *Chin J Electron* 32(3):1–8
31. Li N, Barthe G, Bhargavan K, Butler K, Cash D, Cavallaro L, Chen H, Chen L, Chen Y, Cortier V, et al (2023). Privacy and security. *ACM Trans* 26(1)
32. Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. In: *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738
33. Manikandan G, Kumar R, Rajesh N, et al. (2023) Image security using visual cryptography. In: *Handbook of Research on Computer Vision and Image Processing in the Deep Learning Era*, pp. 281–292. IGI Global
34. Maurya R, Rao GE, Rajitha B (2022) Visual cryptography for securing medical images using a combination of hyperchaotic-based pixel, bit scrambling, and dna encoding. *Int J Inf Technol* 14(6):3227–3234
35. Mhala NC, Jamal R, Pais AR (2018) Randomised visual secret sharing scheme for grey-scale and colour images. *IET Image Proc* 12(3):422–431
36. 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
37. Mirza M, Osindero S (2014) Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*
38. Paul A, Kandar S, Dhara BC (2022) Boolean operation based lossless threshold secret image sharing. *Multim Tools Appl* 81(24):35293–35316
39. Prashanti G, Bhat MN (2023) Cheating identifiable polynomial based secret sharing scheme for audio and image. *Multim Tools Appl* pp. 1–21
40. Rawat AS, Deshmukh M, Singh M (2023) A novel multi secret image sharing scheme for different dimension secrets. *Multim Tools Appl* pp. 1–37
41. Sarkar P, Nag A, Singh JP (2023) Blockchain-based authenticable (k, n) multi-secret image sharing scheme. *J Electron Imaging* 32(5):053019–053019
42. Shivani S (2018) Multi secret sharing with unexpanded meaningful shares. *Multim Tools Appl* 77:6287–6310
43. Stephen A, Punitha A, Chandrasekar A (2023) Designing self attention-based resnet architecture for rice leaf disease classification. *Neural Comput Appl* 35(9):6737–6751
44. Tulsani H, Chawla P, Gupta R (2017) A novel steganographic model for securing binary images. *Int J Inf Technol* 9:273–280
45. Wallat EM, Wuschner AE, Flakus MJ, Gerard SE, Christensen GE, Reinhardt JM, Bayouth JE (2023) Predicting pulmonary ventilation damage after radiation therapy for nonsmall cell lung cancer using a resnet generative adversarial network. *Medical Physics*
46. Wu B, Xie D, Chen F, Zhu H, Wang X, Zeng Y (2023) Compressed sensing based visually secure multi-secret image encryption-sharing scheme. *Multim Tools Appl* pp. 1–23
47. Xie D, Wu B, Chen F, Wang T, Hu Z, Zhang Y (2023) A low-overhead compressed sensing-driven multi-party secret image sharing scheme. *Multim Syst* pp. 1–16
48. Yang CN, Li P, Kuo HC (2023) (k, n) secret image sharing scheme with privileged set. *J Inform Sec Appl* 73:103413