CSIS: compressed sensing-based enhanced-embedding capacity image steganography scheme

Rohit Agrawal1, Kapil Ahuja1∗
1 Mathematics of Data Science (MODS) Laboratory, Indian Institute of Technology Indore, Simrol, Indore, India
∗ E-mail: kahuja@iiti.ac.in

Abstract: Image steganography plays a vital role in securing secret data by embedding it in the cover images. Usually, these images are communicated in a compressed format. Existing techniques achieve this but have low embedding capacity. Enhancing this capacity causes a deterioration in the visual quality of the stego-image. Hence, our goal here is to enhance the embedding capacity while preserving the visual quality of the stego-image. We also intend to ensure that our scheme is resistant to steganalysis attacks.

This paper proposes a Compressed Sensing Image Steganography (CSIS) scheme to achieve our goal while embedding binary data in images. The novelty of our scheme is the combination of three components in attaining the above-listed goals. First, we use compressed sensing to sparsify cover image block-wise, obtain its linear measurements, and then uniquely select permissible measurements. Further, before embedding the secret data, we encrypt it using the Data Encryption Standard (DES) algorithm, and finally, we embed two bits of encrypted data into each permissible measurement. This is the first attempt to rigorously embed more than one bit. Second, we propose a novel data extraction technique, which is lossless and completely recovers our secret data. Third, for the reconstruction of the stego-image, we use the least absolute shrinkage and selection operator (LASSO) for the resultant optimization problem. This has the advantages of fast convergence and easy implementation. This component is also new.

We perform experiments on several standard grayscale images and a color image, and evaluate embedding capacity, Peak Signal-to-Noise Ratio (PSNR) value, mean Structural Similarity (SSIM) index, Normalized Cross-Correlation (NCC) coefficients, and entropy. We achieve 1.53 times more embedding capacity as compared to the most recent scheme. We obtain an average of 37.92 dB PSNR value, and average values close to 1 for both the mean SSIM index and the NCC coefficients, which are considered good. Moreover, the entropy of cover images and their corresponding stego-images are nearly the same. These assessment metrics show that CSIS substantially outperforms existing similar steganography schemes.

1 Introduction

The primary concern during the transmission of digital data over communication media is that anybody can access this data. Hence, to protect this data from being accessed by illegitimate users, the sender must employ some security mechanisms. In general, there are two main approaches used to protect secret data; cryptography [1] and steganography [2]. In cryptography, the encryption process transforms the secret data, known as plain-text, into cipher-text using an encryption key. This text is in unreadable form, hence, it attracts the opponents to exploit the content of the cipher-text by employing some brute-force attacks [3]. However, steganography avoids this scenario.

Steganography is derived from the Greek words steganos means “covered or secret” and graphie means “writing”. In steganography, the secret data is hidden into some other unsuspected cover media so that it is visually imperceptible. Here, both the secret data as well as the cover media may be text or multimedia. The media obtain after embedding secret data into cover media is called stego-media. Some recent steganography schemes that use text as cover media are [3] and [4]. In [3], the authors have proposed an Arabic text steganography scheme, where the secret message is hidden within the text by using Unicode standard encoding. In [4], the authors have proposed a character-level text generation-based linguistic steganography scheme, where the secret message is embedded in the text’s content.

Recently, the steganography schemes that use images as the cover media have gained a lot of research interest due to their heavy use in Internet-based applications. Typically, these images are transmitted in a compressed format. So here, we focus on compressed domain-based image steganography. In this, the challenges are:

1. Improving the embedding capacity.
2. Maintaining the quality of the stego-image.
3. The scheme should be resistant to steganographic attacks.

Although images can be embedded into images, our focus is on embedding binary data into images.

In the following paragraphs, first we discuss the way in which secret data can be embedded into cover images, then we summarize some existing schemes and their limitations, and finally we argue how the scheme presented in this paper outperforms the existing schemes.

Secret data can be embedded in images by two ways: spatially and by using a transform. In the spatial domain based image steganography scheme, secret data is embedded directly into the image by some modification in the values of the image pixels. Some well-known schemes here are listed in [2][5][11]. In the transform domain based image steganography scheme, first, the image is transformed into frequency components, and then the secret data is embedded into these components. Some commonly used such schemes are JSteg [12], F5 [13], and Outguess [14]. Some other techniques, which do not carry specific names are given in references [15][23].

The spatial domain based image steganography outperforms the transform domain one in terms of embedding capacity, but the stego-image has a high amount of redundant data. Digital images transmitted through communication media are usually of this type. Since transform based schemes reduce the redundancy present in the image and represent it in a compressed form, they are preferred for transmission.

Most of the transform domain based scheme follow either Discrete Cosine Transform (DCT) or Wavelet Transform (WT). The DCT based schemes are also called the JPEG compression based image steganography techniques. Several variants of DCT based schemes have been proposed in the literature [12][17][19][23]. For
The scheme that we propose satisfies all the goals mentioned in the earlier paragraphs, i.e. increased embedding capacity without degrading the quality of stego-images as well as making the scheme resistant to steganalysis attacks. Our scheme has three components, which we discussed next. The first component of our scheme consists of three parts; (i) we use compressed sensing to sparsely cover image block-wise and obtain linear measurements. Here, we design an adaptive measurement matrix instead of using a random one. Using our adaptive measurement matrix, we uniquely select a large number of permissible measurements compared to existing schemes. Hence, we achieve a high embedding capacity. Moreover, these measurements act as encoded transformed coefficients, and hence, this adds security to our proposed scheme as well; (ii) we encrypt the secret data using the Data Encryption Standard (DES) algorithm [4]. This adds another layer of security to our scheme; (iii) we embed two bits of secret data into each permissible measurement instead of commonly embedding one bit per measurement. This is a first attempt to rigorously embed more than one bit. Second, we completely extract secret data without any loss using our extraction algorithm. Third, we use the alternating direction method of multipliers (ADMM) solution of the least absolute shrinkage and selection operator (LASSO) formulation of the underdetermined optimization problem in the stego-image construction. The advantages of using ADMM and LASSO are that they have broad applicability in the domain of image recovery and have objective function’s property, have fast convergence, and are easy to implement. This is also a completely new contribution.

For performance evaluation, we perform experiments on standard test images. To check the quality of stego-image, we reconstruct it from the obtained modified measurements and then compare it with its corresponding cover image. We evaluate embedding capacity, Peak Signal-to-Noise Ratio (PSNR) value, mean Structural Similarity (SSIM) index, Normalized Cross-Correlation (NCC) coefficient, and entropy. We achieve 1.53 times more embedding capacity when compared with the most recent scheme of this category. We achieve a maximum of 40.86 dB and an average of 37.92 dB PSNR values, which are considered good. The average values of mean SSIM index and NCC coefficients are close to 1, which are again considered good. Moreover, the entropy of cover images and their corresponding stego-images are nearly the same. In the Experimental Results section, we also show that our scheme outperforms existing compression based steganography schemes [6, 12–14, 16–19].

The rest of the paper has four more sections. Section 2 describes the compressed sensing technique. Section 3 explains our proposed steganography scheme including embedding of the data, extracting it, and stego-image reconstruction process. Section 4 presents the experimental results. Finally, Section 5 gives conclusions and future work.

2 Compressed Sensing

Compressed sensing is used to acquire and reconstruct the signal efficiently. Traditionally, the successful reconstruction of the signal from the measured signal must follow the popular Nyquist/Shannon sampling theorem, which states that the sampling rate must be at least twice the signal bandwidth. In many applications, such as image, audio, video, data mining, and wireless communications & networks, where the signal is sparse or sparsified in some domain, the Nyquist rate is too high to achieve. There is a fairly new paradigm, called compressed sensing that can represent the sparse signal by using a sampling rate significantly lower than the Nyquist sampling rate [22, 33]. Hence, the application of compressed sensing has gained popularity in many areas. Some of them are image processing [24, 44], radar system [35], MRI Imaging [25], and noise separation from data [26].

Compressed sensing projects the sparse signal onto a small number of linear measurements in such a way that the structure of the signal remains the same. The sparse signal can be reconstructed...
approximately from these measurements by an optimization technique. However, the reconstruction of the signal is possible only when the original signal is sparse, and it satisfies the Restricted Isometric Property (RIP) \([37]\) (discussed in Section 2.2). If the original signal is not sparse, then it can often be artificially sparsified. A brief description of signal sparsification, obtaining linear measurements, and reconstruction of the approximate sparse signal is given next.

2.1 Signal Sparsification

Let the original signal be \(x \in \mathbb{R}^{N \times 1}\). The signal \(x\) is \(K\) sparse when it has maximum \(K\) number of non-zeros coefficients, i.e. \(\|x\|_0 \leq K\), where \(\| \cdot \|_0\) denotes the \(l_0 -\text{norm}\) of a vector, and the remaining coefficients are zero or nearly zero. Let the original signal \(x\) not be sparse and be represented in terms of \(\{\psi_i\}_{i=1}^N\) basis vectors each of length \(N \times 1\), then

\[
s = \Psi^T x,
\]

where \(s \in \mathbb{R}^{N \times 1}\) and \(\Psi = [\psi_1, \psi_2, ..., \psi_N] \in \mathbb{R}^{N \times N}\) is an orthogonal matrix. If \(K \ll N\) then this signal is sparsifiable \([38]\). \(s\) is the sparse representation of \(x\), and \(\Psi\) is the corresponding sparsification matrix.

2.2 Sensing Matrix and Linear Measurements

In the compressed sensing framework, we acquire \(M(M < N)\) linear measurements from the inner product between the original signal \(x \in \mathbb{R}^{N \times 1}\) and \(M\) measurement vectors \(\{\phi_i\}_{i=1}^M\), where \(\phi_i \in \mathbb{R}^{N \times 1}\). Considering the measurement/sensing matrix as \(\Phi = [\phi_1^T; \phi_2^T; \ldots; \phi_M^T] \in \mathbb{R}^{M \times N}\), the measurements \(y \in \mathbb{R}^{M \times 1}\) are given as \([38]\)

\[
y = \Phi x.
\]

If the input signal is not sparse but sparsifiable, then using the above theory we get

\[
y = \Phi \Psi s = \Theta s,
\]

where \(\Theta = \Phi \Psi\) is again the measurement matrix of size \(M \times N\). Usually, in the compressed sensing framework, the measurement matrix is nonadaptive. That is, the measurement matrix is fixed and does not depend on the signal. However, in certain cases, adaptive measurements can lead to significant performance improvement.

The main concern here is to design the measurement matrix in such a way so that the most of the information and the structure of the signal is preserved in the measurements. This would imply that original signal would be recovered efficiently from these measurements. To achieve this, for all \(K\)-sparse signals \(s\), the measurement matrix should hold the following inequality \([37]\):

\[
(1 - \delta_K) \leq \frac{\|\Theta s\|^2}{\|s\|^2} \leq (1 + \delta_K),
\]

where \(\delta_K \in (0, 1)\) is an isometric constant. The above inequality is called the RIP that informally says that the \(l_2 -\text{norm}\) of the sparse signal \(s\) and the measurement \(\Theta s\) should be comparable. Apart from satisfying the RIP, the minimum number of measurements required, i.e. the minimum value of \(M\), is also a concern in the measurement matrix design.

2.3 Reconstruction of the Approximate Signal

As discussed in the previous subsection, size of the measurement \(y = \Phi x = \Phi \Psi s = \Theta s\) is less than the size of the original signal \(s\). Hence, the reconstruction of the signal from measurements becomes an ill-posed problem. That is, the solution of an under-determined linear system of equations is to be found. If the matrix \(\Theta\) satisfies the RIP, then the sparse signal \(s\) can be reconstructed approximately by solving the following optimization problem \([39]\):

\[
\min \{\text{number of } i \text{ such that } s(i) \neq 0\}
\]

Subject to \(\Phi \Psi s = y\).

In the above equation, the function to be minimized is simply the number of nonzero coefficients in the vector \(s\). This equation is referred to as \(l_0 -\text{norm}\) minimization problem. It is combinatorial and an NP-hard problem \([39]\). The other approach is to substitute the \(l_0 -\text{norm}\) by the closest convex norm, i.e. the \(l_1 -\text{norm}\), or

\[
\min_{s} \|s\|_1
\]

Subject to \(\Phi \Psi s = y\),

where \(\| \cdot \|_1\) denotes the \(l_1 -\text{norm}\) of a vector. The approach to reconstruct the sparse signal \(s\) by solving the above equation is termed as a convex optimization method.

Other approaches such as Greedy based (OMP \([40]\), CoSaMP \([41]\)), sparse reconstruction by separable approximation \([42]\), Bayesian strategy \([43]\), and ADMM solution of the LASSO formulation of the above optimization problem can also be used to reconstruct the sparse signal from the measurements \([44, 45]\).

Next, we give a brief idea of LASSO and ADMM, which we use. The general LASSO problem is given as \([45]\)

\[
\min_{z} \left\{ \frac{1}{2} \|Az - b\|_2^2 + \lambda \|z\|_1 \right\},
\]

where \(z \in \mathbb{R}^n\), \(A \in \mathbb{R}^{p \times n}\), \(b \in \mathbb{R}^p\), \(\| \cdot \|_2\) is the \(l_2\) norm and \(\lambda > 0\) is a scalar regularization parameter also called Lagrangian parameter \([46]\). Further, \((7)\) is transformed into a form solvable by ADMM \([44]\). That is

\[
\min_{z,z_1} \left\{ \frac{1}{2} \|Az - b\|_2^2 + \lambda \|z_1\|_1 \right\}
\]

Subject to \(z - z_1 = 0\).

Finally, ADMM solve the above optimization problem.

Now, we discuss how to solve our signal reconstruction problem, i.e. \((6)\) by LASSO and ADMM. For our case, \(\Theta = \Phi \Psi\) is the measurement matrix, and \(\Theta \in \mathbb{R}^{M \times N}\). In the compressed sensing framework, matrix \(\Theta\) is underdetermined, i.e. \(M < N\). Hence, there is equivalent solution of \((6)\), which is given as \([27]\)

\[
\min_{s} \left\{ \frac{1}{2} \|\Theta s - y\|_2^2 + \lambda \|s\|_1 \right\}
\]

Here, we observe that \((9)\) is equivalent to \((7)\) with \(\Theta = A\), \(s = z\) and \(y = b\).

Finally, we briefly mention a theoretical result related to reconstruction. In \([48]\), it is shown that for sufficiently small constant \(C (C > 0)\), the \(K\)-sparse signal \(s\) of size \(N\) can be approximately reconstructed from \(M\) measurements \(y\) if \(M \geq CK \log N\). After recovering the sparse signal \(s\), the original signal \(x\) can be obtained as \(x = \Psi s\). For us, this property holds.

3 Proposed Method

Our proposed compressed sensing-based image steganography scheme consists of the following components; data embedding, data extraction, and stego-image construction, which are discussed in the respective sections below.

3.1 Data Embedding

The first step in any compressed sensing-based image steganography scheme is the input image's sparsification if it is not sparse at the
start. This step is equivalent to the signal sparsification of Section
2.1. Methods such as K-SVD, DCT, Discrete Walsh Transform, Sta-
tionary Wavelet Transform, and Discrete Raja Transform provide
good sparsification. Since the distortion due to DCT is less, we use
it as our sparsifying agent. To further reduce the distortion, instead
of sparsifying the whole image at once, first, we decompose the image
into non-overlapping blocks of the same size, and then each
block is sparsified.

Let the image I’s size be \( r_1 \times r_2 \) and each block size be \( B \times B \),
then we have \( (r_1 \times r_2) / B^2 \) number of blocks. In our case, \( r_1 = r_2 \)
and \( B \) completely divides \( r_1 \). The block-wise sparsification is now
done as

\[
s_i = DCT(x_i),
\]

where \( i = 1, 2, \ldots, (r_1 \times r_2) / B^2 \), \( x_i \) and \( s_i \) are the \( i^{th} \) original
and sparse blocks of the same size, i.e. \( B \times B \), respectively. Next,
we convert each block into their vector representation by stacking
them column-wise. Thus, \( s_i \) becomes a vector of size \( B^2 \times 1 \).
Because of sparsification, each sparse vector has few coefficients of
large values and the remaining coefficients of very small values or
zero. Hence, we categories each vector into two groups. Let \( p_1 \) be
the number of coefficients having large values and \( p_2 \) be the number
of coefficients having small values or zero values. Note that here,
\( p_1 < p_2 \) as each of these vectors are sparse in nature and \( p_1 + p_2 = B^2 \).
We represent each vector in two groups based upon these coefficients, i.e. \( s_{i,u} \in \mathbb{R}^{p_1} \) and \( s_{i,v} \in \mathbb{R}^{p_2} \). Now, we project
each sparse vector onto linear measurements using a measurement
matrix, which is equivalent to Section \( 2.2 \).

There are two ways to choose the measurement matrix: either ran-
domly or deterministically. Randomly generated matrices such as the
Independent and Identically Distributed (i.i.d.) Gaussian matrix, the
Bernoulli matrix or other matrices generated by probabilistic meth-
ods are nonadaptive, although they satisfy the RIP. Deterministically
generated matrices are the ones that are designed such that specific
properties are satisfied, e.g., adaptiveness and the RIP. We design a
deterministic matrix that is adaptive to our sparse vector since this
improves the efficiency of compressed sensing. To achieve RIP here,
the projected linear measurements are enforced to have almost the
same \( \ell_2 - norm \) as that of the sparse vector.

One way to design the measurement matrix is to first analyze the
distribution of all \( B^2 \) coefficients in each sparse vector, and then find
the \( m \) indices out of these that give maximum \( \ell_2 - norm \) \( \text{[59]} \). That
is,

\[
E_{\text{max}}^{[m]} = \max_{1 \leq m \leq B^2} \| s_i \|_2^2,
\]

where \( |m| \) is the number of entries in set \( m \) and \( E_{\text{max}}^{[m]} \) is a vari-
able that stores the maximum value of square of \( \ell_2 \)-norm of vector
\( s_i \) for \( i \in m \subset B^2 \). However, in this paper, we use the property
of DCT to design the measurement matrix. This property states
that DCT coefficients can be divided into three sets: low frequency,
middle frequency, and high frequency components. Low frequency
component corresponds to the overall image information, middle frequency
component corresponds to the structure of the image, and high frequency
component relates to the noise or small variance. For image reconstruction,
only lower and middle frequency components are useful. Hence,
we select \( m \) indices out of all \( B^2 \) indices that correspond to these
two sets of frequency \( \text{[15]} \). Here, \( |m| \) is a user-defined parameter
such that \( p_1 < |m| < p_1 + p_2 \), and is discussed in Experimental
Results section. As discussed earlier, in this subsection we have two
groups of sparse vectors \( s_{i,u} \) and \( s_{i,v} \). Hence, we design two dif-
ferent measurement matrices \( \Phi_u \) and \( \Phi_v \) corresponding to \( s_{i,u} \) and
\( s_{i,v} \), respectively.

Since \( \| s_{i,u} \|_2 \) is close to \( \| s_i \|_2 \) because \( s_{i,u} \) contains large value
coefficients of \( s_i \), we project \( s_{i,u} \) onto the same number of linear
measurements. Thus, we have \( \Phi_u = \alpha I_{p_1} \), where \( I_{p_1} \) is the identity
matrix of size \( p_1 \times p_1 \), and \( \alpha \) is a constant.

As mentioned in Section \( 2.2 \) the main purpose of measurement
matrix is to project the sparse vector onto less number of linear mea-
surements. Hence, we project \( s_{i,u} \) onto \( |m| - p_1 \) measurements or
the size of \( \Phi_u \) is \( (|m| - p_1) \times p_1 \). To construct \( \Phi_u \), we first take a
random Hadamard matrix of size \( p_2 \times p_2 \), which is a standard pro-
cedure in compressed sensing literature \( \text{[50]} \), and then we choose
\( |m| - p_1 \) rows from the available \( p_2 \) rows. These rows map to the
last of \( |m| - p_1 \) indices from the index set \( m \). This is because the
first \( p_1 \) indices have the overall image information, and hence, map
to construction of \( \Phi_u \).

We use the same measurement matrices for all blocks. This is
because, for all blocks of an image, the distribution of coefficients of
the generated sparse vectors is almost the same. Thus, for each block
\( i = 1, 2, \ldots, (r_1 \times r_2) / B^2 \), the block-wise linear measurements
vector \( y_i \in \mathbb{R}^{[m]} \) is given as

\[
y_i = \begin{bmatrix} y_{i,u} \\ y_{i,v} \end{bmatrix} = \begin{bmatrix} \Phi_u s_{i,u} \\ \Phi_v s_{i,v} \end{bmatrix}.
\]

Using the standard terminology \( \text{[32, 33]} \), the measurements \( y_{i,u} \) are
called the ordinary samples or non-compressed samples, and the
measurements \( y_{i,v} \) are called the compressed sensing samples.

Next, we discuss the encryption process of the secret data \( D \) that
is to be embedded. This data is a sequence of 0s and 1s. As men-
tioned in the Introduction, this provides an extra layer of security
Algorithm 1 Embedding Rule

Input:
- y: Sequence of transform coefficients.
- S: Encrypted secret bit sequences which is to be embedded.

Output:
- z: The modified version of transform coefficients.

1: if (length(S) < 2 \times length(y)) then
2:   for j = 1 to length(y) do
3:     if (y(j) = 1 or y(j) = 0 or y(j) = +1) then
4:       z = y \ (Do not embed in these measurements)
5:     else
6:       if (y(j) \equiv 2 = 0) then
7:         z = y + 1
8:       else if (S(j) = 0) then
9:         z = y
10:      else if (S(j) = 1) then
11:        z = y + 1
12:      else if (S(j) = 0) then
13:        z = y
14:      else if (S(j) = 1) then
15:        z = y + 1
16:     end if
17:   end if
18: end for

Algorithm 2 Extraction Rule

1: Bit Error Rate (BER)

Error Bits (EB) = D \oplus D', \quad (14)

BER = \frac{\text{Number of ones in EB}}{\text{Size of D}} \times 100\% \quad (15)

Fig. 2: The Extraction Process

sequence of 0s and 1s \[1\]. DES is a fairly standard algorithm used for data encryption \[1\]. Then, we represent S as a set of two-two bits, i.e. \(S = \{S_1, S_2, \ldots, S_n\}\), where each \(S_L\) consists of two bits.

Next, we embed the secret data in our linear measurements \(y_i\). The embedding rule is summarized in Algorithm 1 and helps to embed two bits into the transform coefficients. The rule is designed in such a way so that the secret data could be extracted without any loss, discussed in Data Extraction and Experimental Result sections.

We embed the data in \(y_{i,v}\) and not \(y_{i,u}\). This is because \(y_{i,u}\) corresponds to sparse vector coefficients of large values, and embedding in it leads to degradation of image quality. Further, in \(y_{i,v}\), the secret data is embedded selectively. We do not embed in \(y_{i,v}\) with measurement value of \(-1, 0\) and 1. This is because our embedding algorithm concatenates the measurement values with integers from \(-3\) to \(+3\), and if these value are \(-1, 0\) or 1, then we may end up getting many 0s after concatenation, which leads to difficulty in the extraction process. After embedding in other measurement values of \(y_{i,v}\), we obtain the modified \(y_{i,v}\), which is termed as \(z_{i,v}\). That is,

\[ z_{i,v} = \begin{cases} y_{i,v} & \text{if } y_{i,v} = -1 \text{ or } 1 \\ y_{i,v} + c & \text{otherwise} \end{cases} \quad (13) \]

where \(c \in \{-3, -2, -1, 0, 1, 2, 3\}\). We obtain our stego-data by concatenating the measurements \(y_{i,u}\) and \(z_{i,v}\) as \(\left[ y_{i,u} \right]_{z_{i,v}}\). The block diagram for this complete data embedding process is given in Fig. 1.

3.2 Data Extraction

In this section, we explain the process of extracting embedded secret data from our stego-data. The steps of this extraction process are given below, which are exactly reverse to our data embedding process.

1. Separate the measurements \(z_{i,v}\) from the stego-data, i.e. \(\left[ y_{i,u} \right]_{z_{i,v}}\), where \(i = 1, 2, \ldots, (r_1 \times r_2)/B^2\) is the block number, and \(u, v\) are indices available from the previous subsection.
2. Extract only those measurements from \(z_{i,v}\), whose values are not equal to \(-1, 0\) or 1. The embedding rule ensures that the embedded data could be extracted without loss. In other words, Algorithm 1 ensures that no secret data is embedded in measurements with values \(-1, 0\) and 1.
3. Extract the encrypted message \(S'\) from the measurements obtained in the above step by applying Algorithm 2.
4. Decrypt this \(S'\) by DES algorithm, and obtain the extracted secret data \(D'\).

Now, we check the correctness of this extracted secret data \(D'\) by comparing it with original secret data \(D\). For this, we use the Bit Error Rate (BER), which is given as \[17\]

\[ \text{Error Bits (EB)} = D \oplus D', \quad (14) \]

\[ \text{BER} = \frac{\text{Number of ones in EB}}{\text{Size of D}} \times 100\%, \quad (15) \]
Algorithm 2 Extraction Rule

Input:
• $z$: Sequence of modified linear measurements. These are $z_{i,u}$ that are not having value equal to 0, 1 or -1. See extraction process in Section 3.2.

Output:
• $S'$: Encrypted secret bit sequences.

1: for $j = 1$ to $\text{length}(z)$ do
2: if $(y(j) = -1$ or $y(j) = 0$ or $y(j) = +1)$ then
3: Continue
4: else if $(z(j) \neq 0)$ then
5: $S'_{(j)} = 01$
6: else if $(z(j) \neq 4)$ then
7: $S'_{(j)} = 0$
8: else if $(z(j) = 11)$ then
9: end if
10: else
11: $S'_{(j)} = 10$
12: end if
13: if $(z(j) - 1) \neq 0)$ then
14: $S'_{(j)} = 0$
15: else if $(z(j) = 10)$ then
16: end if
17: else
18: $S'_{(j)} = 1$
19: end if
20: return $S'$

where $\oplus$ denotes the bitwise XOR/Exclusive OR operation. The BER value for our steganography scheme is 0%, i.e. we successfully extract complete secret data without any error. This is the property of our embedding rule. The above extraction process is represented via a block diagram in Fig. 2.

3.3 Stego-Image Construction

When the stego-data is transferred over a communication media, the intruder can access this data from the public channel and can try to construct the stego-image. If the intruder obtains a high visual quality image, then the goal of steganography is fulfilled. This is because he/she will not be able to judge whether some data is hidden in the image or not. Therefore, in this subsection, we give the steps to construct the stego-image from the stego-data, which is equivalent to Section 2.3. We refer this process as construction rather than reconstruction.

1. Obtain the approximate sparse vector $s'$ from the stego-data and measurement matrices $\Phi_u$ and $\Phi_v$ as (recall (12))

\[ s'_{1,u} = \Phi_u^{-1} y_{1,u}, \text{ and } s'_{1,v} = \text{ADMM-LASSO} \left( z_{i,v}, \Phi_v \right). \]  

(16)

Here, as discussed in Section 2.3, we use ADMM and LASSO to construct $s'_{1,v}$. The sparse vector $s'$ is obtained by concatenating $s'_{1,u}$ and $s'_{1,v}$. Here, the size of $s'_{1,u}, s'_ {1,v},$ and $s'$ is the same as that of $s_{1,u}, s_{1,v},$ and $s$, respectively.

2. Convert each vector $s'_{1}$ into a block of size $B \times B$.

3. Apply two-dimensional Inverse DCT (IDCT) to each of these blocks to generate blocks $x'_i$ of image. That is, recall (10).

\[ x'_i = \text{IDCT} \left( s'_i \right). \]  

(17)

4. Construct the stego-image of size $r_1 \times r_2$ by arranging all these blocks $x'_i$.

The block representation of these steps is given in Fig. 3. We show in the Experimental Results section that image obtained from this stego-data preserves the quality of the original image.

As earlier, we term our proposed steganography scheme as Compressed-Sensing-Image-Steganography (CSIS) because we use compressed sensing to enhance the embedding capacity of the image steganography scheme.

![Fig. 3: Stego-Image Construction](image)

4 Experimental Results

Experiments are carried out in MATLAB on a machine with an Intel Core i3 processor @2.30 GHz and 4GB RAM. We use a set of standard grayscale images to test our CSIS. Sample test images are shown in Fig. 4 and Fig. 5. These images have the varying texture property and are taken from the miscellaneous category of USC-SIPI image database [51] and two other public domain databases [52,53].

The miscellaneous category of USC-SIPI database consists of 24 grayscale images. Some images, such as Lena, and Tiffany are no longer available in this database. These images have played a significant role in image processing, and literature. Thus, we use other public-domain test images databases [52,53] for them. A total of seven such images are chosen. Hence, we have a total of 31 grayscale images. Our CSIS is also applicable to color images, and we pick one of them from USC-SIPI database.

In this manuscript, we report average values of all the 31 images with detailed results for 10 images due to space limitations. This is further justified by the fact that the image processing literature has used these images as a subset of them.

The size of each of test images is $512 \times 512$, i.e. $r_1 \times r_2$. We take blocks of size $8 \times 8$, i.e. $B \times B$. As earlier, the size of measurement matrix $\Phi_u$ is $p_1 \times p_1$. Recall from Section 3.1, $p_1$ is the number of coefficients with large values/low frequency in the input sparse vector. For commonly used images, this value is between 10 and 14 [15, 53]. Since the measurement matrix cannot be different for every input matrix, we do experiments with three different values of $p_1$ ($10, 12, 14$) to find the optimal one here. Again from Section 3.1, the size of measurement matrix $\Phi_v$ is $(|m| - p_1) \times p_2$. We take $p_2$ form the following range [15, 54]: $\{32, 35, 36, 37, 39, 40, 42, 47\}$, and as before, $p_2 = B \times B - p_1$ (i.e. $p_2 = 64 - p_1$). For secret data, we use randomly generated data, which is sequence of 0 and 1 bits.

First, we check the embedding capacity of our proposed scheme. Second, we do the similarity analysis between the cover images and the constructed stego-images by assessing. Third, in the remainder of this section, we do security analysis, perform five comparisons with existing steganography schemes, and also experiment with a color image.

4.1 Embedding Capacity Analysis

Embedding capacity is defined as the maximum number of bits embedded in the cover media, which is the image here. The embedding capacity of our proposed steganography scheme depends on the sampling rate (SR), which is given as

\[ SR = \frac{\text{Total Linear Measurements}}{\text{Total Pixels in Cover Image}}. \]  

(18)

We have $r_1 \times r_2$ total pixels in the cover image and $|m|$ linear measurements for each block with $r_1 \times r_2$ number of blocks. Therefore, our sampling rate is

\[ SR = \left( \frac{|m|}{r_1 \times r_2} \right) \times \left( \frac{r_1 \times r_2}{B \times B} \right) = \frac{|m|}{B \times B}. \]  

(19)
From this definition, it is evident that embedding capacity mainly depends upon $|m|$, however, the compressed image quality depends upon both $p_1$ and $|m|$. Therefore, to maintain the quality of stego-image while enhancing embedding capacity, the combination of these parameters is critical.

For different combinations of $p_1$ and $|m|$, in Table 1 we give the embedding capacity in bits of our proposed CSIS for the 10 test images of Fig. 4 and Fig. 5, and the average capacity for all the 31 images. We analyze the data of this table by comparing $p_1$ and $|m| - p_1$ instead of $p_1$ and $|m|$ because the former set directly maps to the number of ordinary samples and compressed sensing samples, respectively. When $p_1$ is constant, and $|m| - p_1$ is increased, the number of compressed sensing samples increases, where the secret data bits are embedded, leading to increased capacity. For example, consider columns 2 and 3 of Table 1 we can observe that the embedding capacity increases when $p_1$ is constant, i.e. 10 and $|m| - p_1$ is increased from 22 to 25. When $|m| - p_1$ is constant and $p_1$ is increased, the number of compressed sensing samples decrease leading to decreased embedding capacity. For example, consider columns 1 and 2, we observe that embedding capacity decreases when $|m| - p_1$ is constant, i.e. 25 and $p_1$ is increased from 10 to 12.

4.2 Stego-image Quality Assessment

In general, when the embedding capacity increases, the visual quality of stego-image degrades. Hence, with increased embedding capacity, preserving the visual quality of stego-image is also essential. There is no universal metric to judge the quality of stego-image. However, we check the quality of stego-image by examining the similarity between cover images and their corresponding stego-images.

This check is done in two ways. Initially we perform a visual or subjective check. The subjective measure is a good way to assess the quality of stego-image, but it depends on many factors like viewing distance, the display device, the lighting condition, viewer’s vision ability, and viewer’s mood. Therefore, it is necessary to design mathematical models to assess the quality of stego-images, which we discuss next.

4.2.1 Subjective or Visual Measure: Human observers are the final arbiter of image quality. Therefore, the subjective measure is a perfect way of assessing the quality of the images. Here, we construct stego-images corresponding to different test images used in our experiment for different combinations of $p_1$ and $|m|$. This result shows that the stego-images are almost similar to their corresponding cover images. The same is true for their corresponding histograms also. As an example, we present the visual comparison for ‘Pepper’ cover image for one set of parameters; $p_1 = 12$ and $|m| = 37$. Fig. 6 shows the (a) ‘Pepper’ cover image (b) ‘Pepper’ cover image histogram (c) ‘Pepper’ stego-image (d) ‘Pepper’ stego-image histogram. From these figures, we observe that the stego-image is almost similar to its corresponding cover image and their corresponding histograms are also very similar.

We also construct the edge map diagrams for both the cover image and its corresponding stego-image for this same example. These edge maps are shown in Fig. 7a and Fig. 7b, respectively. We can see from these figures that both the edge maps are almost the same. Hence, the visual quality of the cover image and its corresponding stego-image is almost similar.

4.2.2 Objective or Numerical Measures: These measures compare the cover images and their corresponding stego-images based on some numerical criteria that do not require extensive subjective studies. Hence, in recent times, these measures are more commonly used for image quality assessment. These include; Peak Signal-to-Noise Ratio (PSNR), mean Structural Similarity (SSIM) index, Normalized Cross-Correlation (NCC) coefficient, and entropy. We discuss all of them below.

**PSNR**: We compute the PSNR value to evaluate the imperceptibility of stego-images. That is,

$$\text{PSNR} = 10 \log_{10} \frac{R^2}{MSE} \text{ dB},$$

where $MSE$ represents the mean square error between the cover image $I$ and the stego-image $SI$. $R$ is the maximum intensity of pixel, which is 255 for grayscale images, and dB refer to decibel. The PSNR is calculated as

$$MSE = \frac{\sum_{i=1}^{r1} \sum_{j=1}^{r2} (I(i,j) - SI(i,j))^2}{r1 \times r2},$$

where $r1$ and $r2$ represent the row and column numbers of the digital image, respectively, and $I(i,j)$ and $SI(i,j)$ represent the pixel value of the cover image and the constructed stego-image, respectively.

A higher PSNR value indicates the higher imperceptibility of the stego-image. In general, a value higher than 30 dB is considered to be good since human eyes can hardly distinguish the distortion in the stego-image [16][55]. The PSNR values of the stego-images corresponding to 10 test images of Fig. 4 and 5 and average for all 31 images for different combination of $p_1$ and $|m|$ are given in Table 2.

![Fig. 4: Test images used in our experiments](image1)

![Fig. 5: Continued from Fig. 4 test images used in our experiments](image2)
The mean SSIM (MSSIM) index is given as

\[
\text{MSSIM}(I, SI) = \frac{1}{M} \sum_{j=1}^{M} \text{SSIM}(i_j, s_j),
\]

where \( \text{SSIM}(x, y) \) calculates the SSIM index for vectors \( x \) and \( y \), and \( \text{MSSIM}(I, SI) \) calculates the mean SSIM between cover image \( I \) and stego-image \( SI \), i.e., for the overall image quality. Here, \( \mu_x \) is the weighted mean of \( x \), \( \mu_y \) is the weighted mean of \( y \), \( \sigma_x \) is the weighted standard deviation of \( x \), \( \sigma_y \) is the weighted standard deviation of \( y \), \( \sigma_{xy} \) is the weighted covariance between \( x \) and \( y \), \( C_1 \) and \( C_2 \) are arbitrary constants, \( i \) and \( s \) represent the pixel value of the cover image and stego-image, respectively, at the \( j^{th} \) local window, and \( M \) is the number of local windows. We took the values of these parameters according to [56]. The value of the mean SSIM index lies between 0 and 1, where the value 0 indicates that there is no similarity between the two images, and the value 1 indicates that the images are exactly similar.

The mean SSIM index values between the stego-images and their corresponding cover images for different combination of \( p_1 \) and \( |m| \) are given in Table [3]. As earlier, 10 images from 4 and 5 are extensively analyze and average of 31 images is reported. From this table, we observe that all these values are close to 1, which represents that the stego-images are very much similar in structure to their corresponding cover images.

**NCC Coefficient:** Normalized correlation (NC) metric measures the degree of similarity between two images, and when the two images are independent, this correlation is called normalized cross-correlation (NCC) [54]. The NCC coefficient is given as

\[
\text{NCC} = \frac{\sum_{i=1}^{r_1} \sum_{j=1}^{r_2} I(i,j)S(I,j)}{\sqrt{\sum_{i=1}^{r_1} \sum_{j=1}^{r_2} I^2(i,j) \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} S^2(i,j)}},
\]

where \( r_1 \) and \( r_2 \) represent the row and column numbers of the digital image, respectively. \( I(i,j) \) and \( S(I,j) \) represent the pixel value of the cover image and the constructed stego-image, respectively. The value equal to 1 indicates that both the images are exactly similar. For our experiments, the values of NCC are given in Table [4]. The set of images used are same as for PSNR and SSIM. We observe that all these values are close to 1, which means that the stego-images are almost identical to their corresponding cover images.

**Entropy:** In general, entropy is defined as the measure of average uncertainty of a random variable, which here is the average number of bits required to describe the random variable. In the context of an image, it is a statistical measure of randomness that can be used to characterize the texture of the image [57]. For a grayscale image, entropy is given as

\[
\text{Entropy} = -\sum_{i=0}^{255} p_i \log_2 p_i,
\]

where \( p_i \) is the probability of value \( i \) pixel of the image. Table [5] gives the entropy values for the cover images and their corresponding stego-images for different combinations of \( p_1 \) and \( |m| \). The set of images used are same as for PSNR, SSIM, and NCC. From this table, we observe that for all these combinations of \( p_1 \) and \( |m| \), the entropy of the cover images and their corresponding stego-images are almost similar.
4.3 Security Analysis

Since the proposed CSIS is a transform domain based technique and it employs indirect embedding strategy, i.e., it does not follow the LSB flipping method, and hence, it is immune to statistical attacks [24, 58]. Also, CSIS does not lead to the shrinkage effect. That means, after embedding, the nonzero coefficients do not modify to zero value, and hence attacks against PS [58] are not considered.
Moreover, in CSIS, the measurement matrix $\Phi$ is considered as the secret-key, which is shared between the sender and the legitimate receiver. If the eavesdropper intercepts the stego-image by a randomly generated measurement matrix, he cannot not enter the embedding domain without the original secret-key. Hence, we achieve increased security in our proposed system. To justify this, we extract the secret data in two ways, i.e. by using the correct measurement matrix and by using a measurement matrix that is very close to the original one, and obtain the BER (discussed in Section 3.2) between the original secret data the extracted one.

In Fig. 8, we present this BER for earlier discussed 10 cover images, and for the parameter $p_1=12$ and $|m|=37$. In this figure, we see that for the correct secret-key, the BER is 0, and for a tiny difference in the measurement matrix, i.e. wrong secret-key, the BER is very high, which is 35% to 40%. That is, even a small change in the secret-key will lead to an extreme shift in accuracy between the original secret data and the extracted one.

In addition to the above security analysis, we also measure the security by analyzing the distribution of the measurements and their corresponding modified measurements, i.e. after embedding the secret data. For ‘Pepper’ image with parameter $p_1=12$ and $|m|=37$, this distribution of the original measurements and the modified measurements is shown in Fig. 9a and Fig. 9b respectively. The green and blue colors are automatically added by Matlab and do not have any significance here. From these figures, we see that the distribution for both cases is almost the same. We also check these distributions for all the images and obtain the same results. We do not include these in this manuscript due to space limitations.

The preservation of distribution of measurements in the earlier two histograms can also be justified by the probability of addition and subtraction operation decided by our algorithm. In Fig. 10, we plot this probability. From this figure, we see that the lines of probabilities of addition and subtraction operation oscillate around 0.5.

Here, the minimum and maximum deviation to 0.5 are 0.02 and 0.07, respectively, i.e. for proposed CSIS, the probabilities of both the addition and the subtraction are nearly the same. The distribution of measurements and the probability of addition & subtraction operation as discussed have justified that for our proposed CSIS, the likelihood of detecting data embedding by an eavesdropper is significantly low.

Fig. 8: BER with the correct and with a wrong secret-key (i.e. measurement matrix)

![Fig. 8: BER with the correct and with a wrong secret-key (i.e. measurement matrix)](image.png)

Fig. 9: Distribution of measurements for ‘Peppers’ image

![Fig. 9: Distribution of measurements for ‘Peppers’ image](image.png)

In this subsection, we compare the performance of the proposed CSIS with the existing steganography schemes. This result is given in Table 6. In this table, the first column represents the comparison metrics, and the remaining columns give the metric data for different steganography schemes.

In the first row of Table 6, we compare the average embedding capacity over all the 31 images. We report these embedding capacity for the parameter $p_1=12$ & $|m|=37$. In this table, we do not compare these results for all the images because the existing schemes’ data are not available for all the images. From the first row of this table, we observe that on an average our steganography scheme has approximately 0.67, 4.23, 4.21, 8.46, 3.18, 183.87, 3.03 and 1.53 times embedding capacity as compared to references [6], [12], [13], [14], [16], [17], [18], and [19], respectively. Here, we can see that our proposed scheme has a higher embedding capacity compared to all schemes except the one, which is [6]. The reason for this is that this scheme is based on embedding secret data in the spatial domain. As discussed in the Introduction, spatial domain based embedding techniques have a higher embedding capacity, but they are prone to security issues. Also, these techniques are not based on compression, which is the main motivation of this manuscript. Further, as evident from Table 7 for a set of parameters $p_1=12$ and $|m|=47$, CSIS has 270937, and 251989 bits embedding capacity for the average of 10 and 31 images, respectively. Hence, for this set of parameters, CSIS has approximately the same embedding capacity as that of [6].

In the second row of this table, for our scheme we report the range of PSNR values when considering all sets of parameters and again all 31 images. From the second row of this table, we observe that similar to existing steganography schemes, our CSIS also has PSNR values greater than 30 dB, which is considered good [16, 53].

The purpose of the proposed CSIS is to embed secret data in the compressed domain. Hence, in the third row of Table 6, we check which schemes are based on compression and which are not. From this row, we observe that except [6] [18], our CSIS and all other schemes are based on compression. Finally, from the fourth row to the sixth row of Table 6, we compare the security of these schemes by checking whether they are resistant to chi-square attack or not, resistant to shrinkage effect or not, and use any secret-key or not. We observe that only our proposed CSIS and [17] schemes pass all the three security tests. Hence, we can conclude that out of all these schemes, only CSIS fulfills all the goals of steganography with higher embedding capacity.

4.4 Performance Comparison

In this subsection, we compare the performance of the proposed CSIS with the existing steganography schemes. This result is given in Table 6. In this table, the first column represents the comparison metrics, and the remaining columns give the metric data for different steganography schemes.

In the first row of Table 6, we compare the average embedding capacity over all the 31 images. We report these embedding capacity for the parameter $p_1=12$ & $|m|=37$. In this table, we do not compare these results for all the images because the existing schemes’ data are not available for all the images. From the first row of this table, we observe that on an average our steganography scheme has approximately 0.67, 4.23, 4.21, 8.46, 3.18, 183.87, 3.03 and 1.53 times embedding capacity as compared to references [6], [12], [13], [14], [16], [17], [18], and [19], respectively. Here, we can see that our proposed scheme has a higher embedding capacity compared to all schemes except the one, which is [6]. The reason for this is that this scheme is based on embedding secret data in the spatial domain. As discussed in the Introduction, spatial domain based embedding techniques have a higher embedding capacity, but they are prone to security issues. Also, these techniques are not based on compression, which is the main motivation of this manuscript. Further, as evident from Table 7 for a set of parameters $p_1=12$ and $|m|=47$, CSIS has 270937, and 251989 bits embedding capacity for the average of 10 and 31 images, respectively. Hence, for this set of parameters, CSIS has approximately the same embedding capacity as that of [6].

In the second row of this table, for our scheme we report the range of PSNR values when considering all sets of parameters and again all 31 images. From the second row of this table, we observe that similar to existing steganography schemes, our CSIS also has PSNR values greater than 30 dB, which is considered good [16, 53].

The purpose of the proposed CSIS is to embed secret data in the compressed domain. Hence, in the third row of Table 6, we check which schemes are based on compression and which are not. From this row, we observe that except [6] [18], our CSIS and all other schemes are based on compression. Finally, from the fourth row to the sixth row of Table 6, we compare the security of these schemes by checking whether they are resistant to chi-square attack or not, resistant to shrinkage effect or not, and use any secret-key or not. We observe that only our proposed CSIS and [17] schemes pass all the three security tests. Hence, we can conclude that out of all these schemes, only CSIS fulfills all the goals of steganography with higher embedding capacity.

4.5 Experiments on Color Image

All the above experiments were performed on the grayscale images. However, we also show the applicability of our proposed CSIS on a
color image. For this we only use ‘Pepper’ color image of resolution 512 × 512, and perform experiments for $p_1 = 12$ and $|n| = 37$ as well as $p_1 = 14$ and $|n| = 36$.

Fig. 1 shows the subjective/ visual measure for ‘Pepper’ color image for $p_1 = 12$, $|n| = 37$. From this figure, we observe that the cover image and its corresponding stego-image are almost similar. Table 7 gives the results for other measures like embedding capacity, PSNR values, mean SSIM index, NCC coefficients for the different color components, and entropy for both cover image and stego-image. We can observe from this table that the embedding capacity of our color image is approximately three times the embedding capacity of ‘Pepper’ grayscale image for the same set of parameters. Please see columns 4 and 8 of Table 7. This is because of the presence of three color components in the color image. Also, the PSNR values here are greater than 30 dB, and mean SSIM index & NCC coefficients are all close to 1, which shows that the stego-image is almost similar to its corresponding cover images. Finally, we compare the entropy of the cover image and the stego-image. We see that entropy for both these images is almost the same.

5 Conclusions and Future Work

We present an enhanced-embedding capacity image steganography scheme based on compressed sensing technique. Here, we combine three components to achieve increased embedding capacity without degrading the quality of stego-images, as well as making it resistant to steganalysis attacks. First, we use compressed sensing to sparsify cover image block-wise and obtain its linear measurements using a matrix. We uniquely select a large number of permissible measurements. Hence, we achieve a high embedding capacity. Since the measurement matrix is a secret-key that is shared between the sender and the legitimate receiver, this adds extra security to our scheme. Also, we encrypt the secret data using the DES algorithm and then embed two bits of secret data into each permissible measurement instead of embedding one bit per measurement. Second, we propose a technique of data extraction that is lossless and recoverable from the stego-image. Third, we use ADMM solution of the LASSO formulation of the obtained optimization problem in the stego-image construction. The reason for selecting them is that they have broad applicability in the field of image processing, require less assumptions on the property of the objective function, have fast convergence, and are easy to implement.

We initially perform experiments on several standard grayscale images that vary in texture, and with different sets of parameters and randomly generated binary data as our secret data. For performance evaluation, we calculate embedding capacity, PSNR value, mean SSIM index, NCC coefficient, and entropy. Experiments show that our proposed CSIS achieves higher embedding capacity than existing steganography schemes that follow compression. We achieve 1.53 times more embedding capacity as compared to the most recent scheme of the similar category. PSNR values coming out of our scheme are more than 30 dB, which is considered good. Both mean SSIM index and NCC coefficients values are close to one, which shows that the cover images and their corresponding stego-images are almost similar. This similarity is further supported by the fact that we obtain approximately the same entropy value for both the cover images and their corresponding stego-images. Further, we also show the applicability of CSIS on a color image. Again, the results obtained are almost the same as that of grayscale images. However, we get approximately three times higher embedding capacity for the color image because of the presence of the three component in color images.

In future, we plan to embed the secret data in text, audio, and video. Other future works include extending this work for a real-time application such as hiding fingerprint data, iris data, medical information of patients, and personal signature. As mentioned in the Introduction, another line of work is embedding images inside images. Since a lot of work has been done in embedding a single image, we will focus on hiding multiple secret images and multilevel image steganography scheme.

### Table 6 Performance comparison between proposed CSIS and various other steganography schemes

| Metrics                  | CSIS        | Ref. [9] | Ref. [12] | Ref. [13] | Ref. [14] | Ref. [18] | Ref. [17] | Ref. [19] |
|--------------------------|-------------|----------|----------|----------|----------|----------|----------|----------|
| Capacity (in bits)       | 174678      | 262144   | 41287    | 41451    | 20644    | 55001    | 950      | 57568    | 113960   |
| PSNR (in dB)             | 30.94 to 40.86 | 36.51    | 34.39    | 35.00    | 34.54    | 35.54    | 35.52    | 49.89    | 36.64    |
| Compression              | Yes         | Yes      | Yes      | Yes      | Yes      | Yes      | Yes      | No       |
| Based to Chi-square      | Yes         | No       | No       | Yes      | Yes      | Yes      | Yes      |
| Resistant to Shrinkage   | Yes         | NA       | Yes      | Yes      | Yes      | Yes      | Yes      |
| Metrics                  | Yes         | No       | No       | No       | No       | No       | No       |

6 References

1. W. Stallings, Cryptography and network security: principles and practice. Prentice Hall, 2019.
2. M. Khodaei and K. Faez, “New adaptive steganographic method using least-significant-bit substitution and pixel-value differing,” IET Image Processing, vol. 6, no. 6, pp. 677–686, 2012.
3. N. Alanazi, E. Khan, and A. Gutub, “Efficient security and capacity techniques for arabic text steganography via engaging unicode standard encoding,” Multimedia Tools and Applications, Accessed on 22 October 2019. [Online].
4. L. Xiang, S. Yang, Y. Liu, Q. Li, and C. Zhu, “Novel linguistic steganography based on character-level text generation,” Mathematics, vol. 8, no. 9, p. 1558, 2020.
5. S. Sarreshtedani and M. A. Akhaee, “One-third probability embedding: a new ±1 histogram compensating image loss significant bit steganography scheme,” IET image processing, vol. 8, no. 2, pp. 78–89, 2013.
6. S. Devi, M. N. Sahoo, K. Muhammed, W. Ding, and S. Bakshe, “Hiding medical information in brain MR images without affecting accuracy of classifying pathological brain,” Future Generation Computer Systems, vol. 99, pp. 235–246, 2019.
7. S. Khan, T. Khan, T. Mahmood, and N. Ahmad, “Analysis of data hiding in R, G and B channels of color image using various number of LSBs,” in 2016 Sixth International Conference on Innovating Computer Technology (INTECH), pp. 270–274, IEEE, 2016.
8. A. Ul Islam, F. Khalid, M. Shah, Z. Khan, T. Mahmood, A. Khan, U. Ali, and N. Anees, “An improved image steganography technique based on MSB using bit differing,” in 2016 Sixth International Conference on Innovating Computer Technology (INTECH), pp. 265–269, IEEE, 2016.
9. W. Zhang, Z. Zhang, L. Zhang, H. Li, and N. Yu, “Decomposing joint distortion for adaptive steganography,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 10, pp. 2274–2280, 2017.
10. A. Rehman, T. Saba, T. Mahmood, Z. Mehmood, M. Shah, and A. Anjum, “Data hiding technique in steganography for information security using number theory,” Journal of Information Science, vol. 45, no. 6, pp. 767–778, 2016.
11. M. Shah, W. Zhang, H. Hu, H. Zhou, and T. Mahmood, “Homomorphic encryption-based reversible data hiding for 3D mesh models,” Arabian Journal for Science and Engineering, vol. 43, pp. 8145–8157, 2018.
12. [Blog Source. Last accessed on 22 October 2019.](https://zoom.in/
paul/crypto/jsteg/)
13. A. Westfeld, “FS—a steganographic algorithm,” in Information Hiding (I. S. Reed, ed.), vol. 2137, pp. 389–392, Springer Berlin Heidelberg, 2001.
14. N. Provos, “Defending against statistical steganalysis,” in Proceedings of the 10th Conference on USENIX Security Symposium, vol. 10 of SSYM’01, pp. 323–336, USENIX Association, 2001.
15. C.-C. Chang, T.-S. Chen, and L.-Z. Chung, “A steganographic method based upon JPEG and quantization table modification,” Information Sciences, vol. 141, no. 1, pp. 125–138, 2002.
16. C.-J. Liu and S.-R. Liao, “High-performance JPEG steganography using complementary embedding strategy,” Pattern Recognition, vol. 41, no. 9, pp. 2945–2955, 2008.
17. J.-S. Pan, W. Li, C.-S. Yang, and L.-J. Yan, “Image steganography based on sub-sampling and compressive sensing,” Multimedia Tools and Applications, vol. 74, no. 21, pp. 9191–9205, 2015.
Table 7  The performance analysis of our proposed scheme on color cover image (512 × 512 Pepper color image) using different parameters.

| Parameters | Embedding capacity | PSNR | Mean SSIM | Normalized cross-correlation (NCC) (for different color component) | Entropy | Cover image | Stego image |
|-----------|-------------------|------|-----------|-------------------------------------------------|---------|-------------|-------------|
|            | $p_1 = 12$, [$m]$ = 37 |      |           | Red | Green | Blue | Average |             |           |
| $p_1 = 14$, [$m]$ = 36 | 503863 | 33.89 | 0.9913 | 0.9990 | 0.9981 | 0.9963 | 0.9978 | 7.669 | 7.723 |
| Average   | 538760 | 33.80 | 0.9878 | 0.9990 | 0.9980 | 0.9962 | 0.9978 | 7.669 | 7.723 |

Fig. 11: 512 × 512 ‘Pepper’ color cover image and stego-image using parameter $p_1=12$, [$m$]=37.