INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

# Hybrid Encoding Schemes in Image Steganography Combined with Scrambling for Safeguarding Legal Asset Documents

**C Yamini**[1*], **N Priya**[2]

**1** Research Scholar, Research Department of Computer Science, SDNB Vaishnav College for Women, Chromepet, Chennai, Tamil Nadu, India
**2** Associate Professor, PG Department of Computer Science, SDNB Vaishnav College for Women, Chromepet, Chennai, Tamil Nadu, India

OPEN ACCESS

**Received:** 06-02-2024

**Accepted:** 01-04-2024

**Published:** 23-04-2024

**Citation:** Yamini C, Priya N (2024) Hybrid Encoding Schemes in Image Steganography Combined with Scrambling for Safeguarding Legal Asset Documents. Indian Journal of Science and Technology 17(17): 1776-1784. https://doi.org/10.17485/IJST/v17i17.342

* **Corresponding author**.

chivkulayamini@gmail.com

**Funding:** None

**Competing Interests:** None



Published By Indian Society for Education and Environment (iSee)

**ISSN**
Print: 0974-6846
Electronic: 0974-5645

## Abstract

**Objective**: The main objective of the study is to provide security to legal asset documents by including the details of ownership of the asset. **Methods**: This paper aims to improve security of legal asset documents using hybrid encoding schemes in Image Steganography combined with Image Scrambling. The proposed work uses images were gathered from various online sources grouped into a dataset of 40 images along with their 40 text files respectively. The techniques that were applied in the research are pre-processing for the noise removal, Steganography to embed data into image and Scrambling to introduce blur to the image. Finally, the objective metrics like Mean-Square Error (MSE), Peak Signal - Noise Ratio (PSNR) and Signal to Noise Ratio (SNR) values were calculated from them and compared with the acceptable values of the respective metrics. **Findings**: In the existing study, Steganography and Scrambling has been used separately for data and image security but in the proposed technique, Steganography has been applied with the land ownership data being hidden in the document image and the image being scrambled using different masks. The objective metrics gave the results such as the MSE value ranged between 0.0 and 10.0; the SNR value ranged between 14 dB to 80 dB; the PSNR value lies between 27 dB to 100 dB. The outcome of these values was mostly within the acceptable value range for all the measures here. **Novelty**: The proposed output dataset contains land asset documents that have details of ownership embedded in itself. Also, the hybrid encoding schemes of both the steganography and scrambling used here gives us data and image security together as proved in the outcome.

**Keywords:** Image Steganography; Scrambling; Text Hiding; Legal Documents; Data Security

## 1 Introduction

Any document that has details regarding the ownership of some asset would mean a lot of value. These documents need to be safeguarded. There are details corresponding to the assets like the owner's information along with the history of the document. Such data can be stored along with the image itself. For that purpose, this paper deals with the combining of Image Steganography to hide the text data and Image Scrambling to secure the overall image. The previous works in this domain would be [1] where the author discusses the various concepts related to Steganography and its pros and cons. In [2] the author describes about how to work in Steganography along with the deep learning methods. The use of Deep learning methods are common nowadays but the process itself becomes tedious. The concept of Hidden Image Encryption and Decryption (HIED) along with the formulation of Mean-Square Error (MSE), Peak Signal - Noise Ratio (PSNR) and Signal to Noise Ratio (SNR) values was used in [3] where the output is achieved through Steganography and data is hidden. But the Concept of image security is not discussed in this. The process of Scrambling is used in [4] whereas no text is hidden inside. This is only half of this paper as the other part is data embedding into the image. Steganography is further discussed with the optimization techniques in [5], but they are related more to the hiding of data alone. In [6], the author elaborates about hiding image inside another image to secure that which may be considered but here the topic is about data security in image along with the image. The author of [7] proposes a novel method on hiding data in image and also keeping the image clarity the same, using Compressive Sensing (CS) and Directional Lifting Wavelet Transform (DLWT). The way this paper differs from the specified paper is that the concept discussed below corresponds to securing the image by scrambling them as well. Quantum Steganography is discussed in [8], where the topic deviates more to the path of physics and the various other methods that can be used to embed data into an image. The paper titled " W-VDSR: wavelet-based secure image transmission using machine learning VDSR neural network" referenced in [9] talks about storing data in a block of image and then securing them. Both of the paper [8] and [9] does not talk about securing the image in return. They talk about maintaining the clarity of the image. Sharma, Vijay & Sharma, et.al referenced in [10] talks about the use of Steganography and then Scrambling them, where the scrambled image is then used in steganography using graph wavelet transform method. Here both the image and data security are achieved in existing authors. This paper focuses on the same, but in an alternate way such that the ownership data is embedded with the hybrid encoding schemes in steganography and the various masks for the image that can be created through scrambling provide us more security for legal asset documents. In the paper referenced at [11] the author Abdulla, Alan., states the different ways through which Steganography is optimized. Either done to receive embedding efficiency or for the distortion function that relates to the statistical detections in the image. They also recommend that authors try to improve the embedding efficiency rather than focusing on the distortion function and the same is done is this paper as for Steganography, the data is embedded totally, and the security of that data is discussed here while keeping the image clarity nearly same.

## 2 Methodology

Legal Asset documents as far as India is concerned are "Patta" and "Chitta" documents. These were maintained in the form of Hard-copy or in Paper. Safeguarding these papers were termed as a tedious process, due to which the Indian Government took the measures to make them available digitally. Nowadays, Patta and Chitta documents can be downloaded through online means by any person with a valid identity in India.

The dataset for this process consists of real-time land asset document images. These images have some kind of noise in them. As the whole process is done step by step manually, a sample dataset of 40 images was used, and they gave the same accuracy comparatively in the results. The text file needed for the embedding was also made-up manually and were used for all the 40 images. That means, each and every image has a separate text document that needs to be embedded into it. Once the output is received, the SNR and PSNR has been calculated, and they were plotted in a graph to show the relation between the noise present in the image with the SNR and PSNR values obtained through the research.

Since, this research also speaks about image security along with data hiding, the amount of blur received would also prove the point that image security has been achieved. The results also show us the decreased PSNR values after Scrambling has been achieved successfully.

### 2.1 Research Flow:

Image Steganography combined with scrambling technique is done here in 6 steps.

1. Image and text File received, and size checking is done - Get the image file and text file and check for length error. If any, stop process and throw error. If no error, move to next process of formulating and_mask and or_mask.
2. The and_mask and or_mask are formulated according to the received file.

3. The encoding process is done using the and_mask and or_mask received in step 2 - Perform bitwise AND first and then bitwise OR on the image file data. That would give the result.
4. The encoded image is then sent for scrambling which in turn would make the image unrecognizable. This needs to be descrambled first to view the clear image and then decoded - The result is then sent for scrambling the image that would give us the output image. That image would be scrambled to beyond recognition. That would mean that only when we descramble that, can we get the clear image. This is then sent for decoding.
5. The decoding of the encoded image to receive original document is done - Decoding goes through the reverse process as that of encoding but the change is input image and output received is text file.
6. The MSE, SNR and PSNR values are then calculated using the respective formula and are plotted in a graph to show the proportionality between them.

### *2.1.1 STEP 1*

The concept here is that the image and data file are taken and checked for the file size. If any one of them like the data file exceeds the size, the user is notified immediately.

The file that has to be encoded needs to be present in the same directory as that of the image that is taken. The output is also stored in the same directory. The code when run using a console needs to be given using the codes specified. It means that the codes specified in Table 1 are used to represent input, output, image file, text file, output file to be stored.

**Table 1. The code to be used for executing the file in the console**

| Code | Stands for |
|---|---|
| -e | Encoding to be done |
| -d | Decoding to be done |
| -i | The image file/path that needs to be encoded/decoded |
| -o | The output file/path where the output image/file needs to be stored |
| -f | The file that is needed to either encode or decode into. |

Using the code mentioned in Table 1, when the image file is encoded with the text, the message is given in the console as 'Image encoded'. If not, the respective error message is given as shown in Figure 1 where the file that needs to be encoded is missing in the directory. Figure 2 is the original image that was taken for encryption. Figure 3 gives us the snip of the file that is holding the data that needs to be embedded into the image.

- **Conversion of data file from bytes to arrays**

```
file_bytes = open(file_path, ”rb”).read()
    return bytes2array(file_bytes)
```

This method is used when the file is being read for the process.

There is another process when the file is being written back after decoding the image.

- **Conversion of data file from arrays to bytes**

```
bytes_data = array2bytes(file_bit_array)
    f = open(file_path, 'wb')
    f.write(bytes_data)
    f.close()
```

### *2.1.2 STEP 2*

Here, the data file is used as the or_mask and then and_mask is formulated from the or_mask received. It includes getting an array full of zeroes with the same shape and size of the or_mask and adding the or_mask itself to that.

### *2.1.3 STEP 3*

Then the encoding process starts with the bitwise and, or masks which result in the data getting encoded into the image. While the previous papers performed the bitwise AND operations on the image file using or_mask; then performed bitwise OR operations on them using the and_mask. This paper focuses on performing the different way around as to first with the

and_mask bitwise AND is performed and or_mask with the result that was received. Once the encryption is done and the successful message is received, the encoded image is saved with the specified file name in the same directory as of the original image. Figure 4 shows the encoded image that is same as the original image in Figure 3, but it has the text in it.

### *2.1.4 STEP 4*

The encoded image is then passed through the scrambling technique, where the image is passed along with the mask that would scramble the image to such an extent that it becomes unrecognizable. This would be needed to first unscrambled to actually get the image that is clear to naked eyes.

```
def createImageInfo(image):
data = []
loByte = image.width & 255
hiByte = image.width >> 8
data.append(hiByte)
data.append(loByte)
loByte = image.height & 255
hiByte = image.height >> 8
data.append(hiByte)
data.append(loByte)
return data
```

This code is used to create information of the image to process that. That would be needed for the new masked image that is being created.

```
def countBlocksOfMask(maskimage):
numberOfBlocks = 0
for y in range(maskimage.height):
  for x in range(maskimage.width):
    luma = maskimage.getpixel((x,y))
    if luma > 0:
    numberOfBlocks = numberOfBlocks + 1
return numberOfBlocks
```

This code is used to get the dimensions of the mask that is being applied to the image. This mask can be created elsewhere and sent along with the image when executing the scrambling part.

### *2.1.5 STEP 5*

The descrambled image is then sent for decoding. The decoding also works on the reverse of the said process except for the flow where instead of image output, we receive only the text that was encoded into the image. Here, each and every time the file is decoded, a new file is opened for that purpose alone. This file is opened with the given file name. If the given name is already present, the file is not overwritten and is left as such. The Figure 6 gives us the snippet of the decoded file with all information intact.

### *2.1.6 STEP 6*

The PSNR and SNR values are calculated using the formula given below. The PSNR value between 30db to 50db is considered as acceptable for any processed image. All the result values received for the samples used here are between the said acceptable values.

Hence, this proves that the quality of the processed image is good as well as the objective of blurred image output after Scrambling is also seen through the human eye itself.

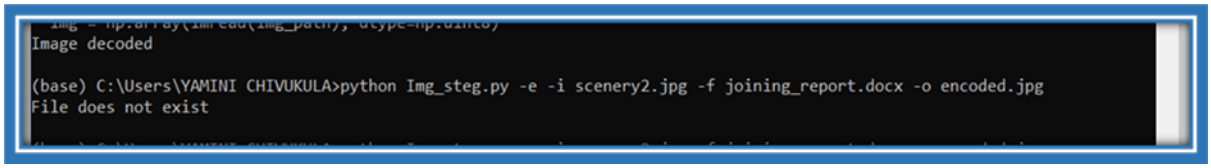
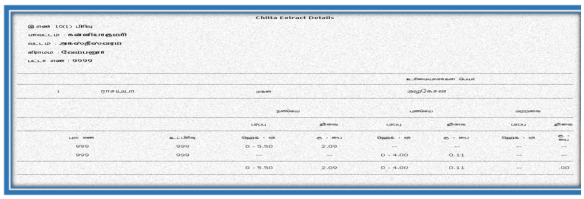
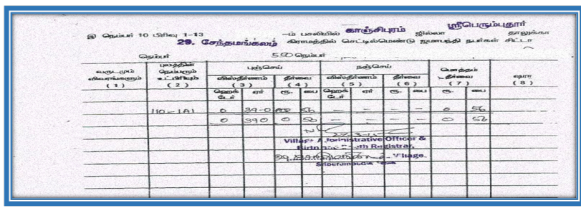



**Fig 1. File not found error is given when the file that needs to be encoded is missing in the directory**

Sample 1



Sample 2



Sample 3

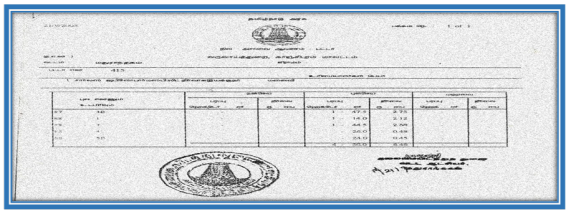
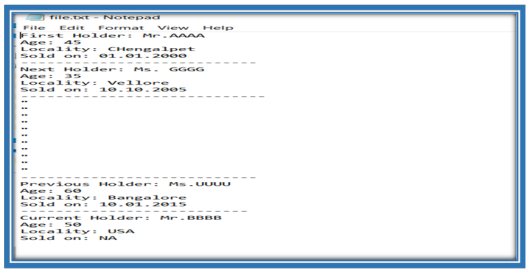


**Fig 2. The original images that was taken for encryption**

Text Sample-1:




Text Sample-2:

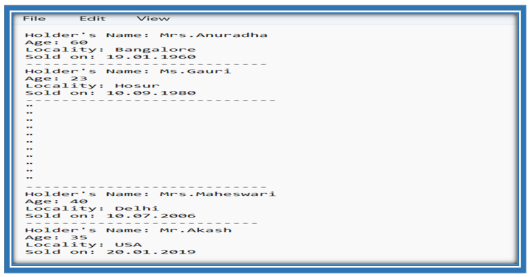



**Fig 3. This gives us the sample snips of the text data that needs to be embedded into the image**

## 3 Results and Discussions

The result here is split into two parts. One being the result from Steganography which is then sent again into scrambling part. The first result is that of the encoding process which gives us the same image without any difference from the original image. This would mean that the data is safely encoded into the image but the image is still visible clearly. In places where the image

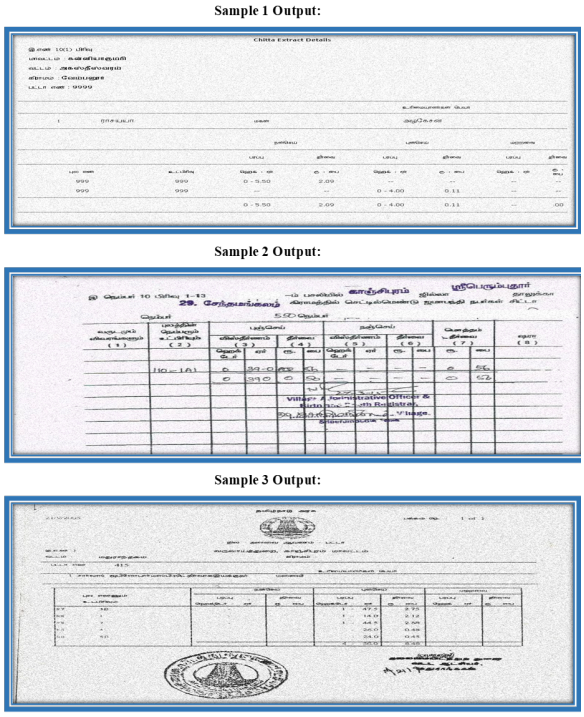



**Fig 4. This shows the encoded images that is same as the original image in Image 2, but it has the text embedded in it**

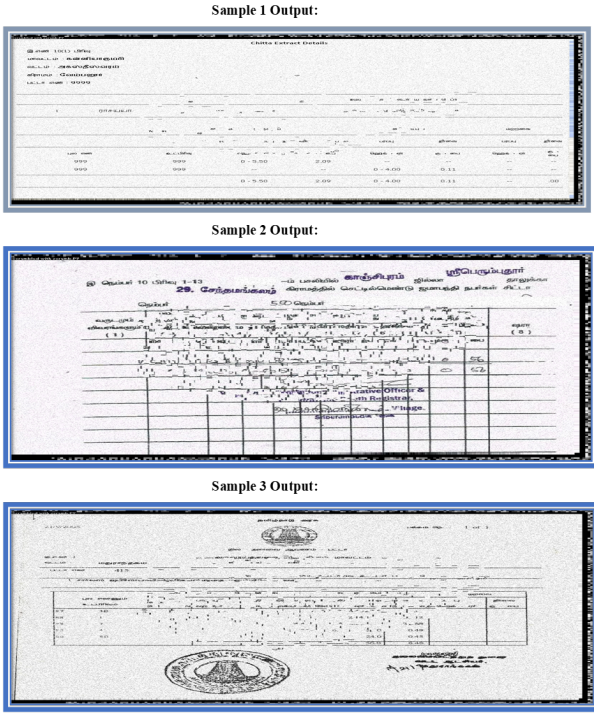



**Fig 5. The output after scrambling the encoded image. The result of these scrambled images needs to be unscrambled first to get the old encoded image which is then processed further**

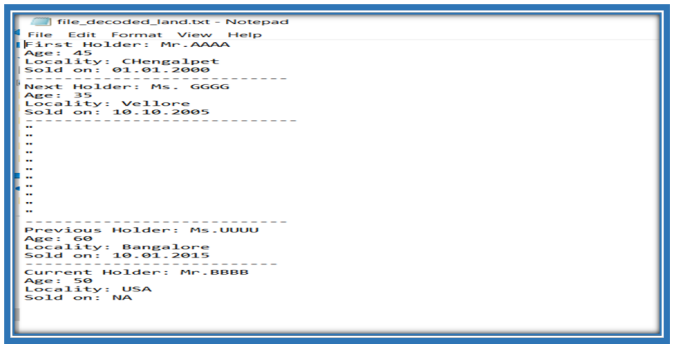



**Fig 6. The output from the decoded text file that is saved after the execution in the console**

needs to be safeguarded, scrambling is also combined and used. When we scramble, we get the image of the same size but not understandable. This would keep the image also safe along with the text data. This is shown in Figure 5. The results were validated and the outcome if it favors or contradicts the existing techniques are discussed in Table 2.

**Table 2. Comparison of this paper's outcome with the existing ones**

| Reference No. | Summary of the existing paper | The outcome as specified by the author | Comparative outline of this paper with the existing paper |
|---|---|---|---|
| (1) | The author discusses about the various concepts related to Steganography and its pros and cons. | The data alone can be secured whereas the image remains with the same clarity. | When compared, Image security is not discussed. Hence the process discussed above would yield more security to the data as well as the image. |
| (2) | The topic here is about how to work in Steganography along with the deep learning methods. | Deep learning techniques are a tedious process and takes time. This also involves only data security. | Data security is comparatively better here but image security is not provided to that extent. |
| (3) | The concept of Hidden Image Encryption and Decryption (HIED) was used with Steganography. PSNR, SNR and MSE values are calculated. | The text is hidden in the Image but the image remains the same. High PSNR values have been received in results. | Compared to this paper, the output of Steganography is the same clarity since the result gives us acceptable values of PSNR similar to this paper, but scrambling the image is a technique that is discussed more in this paper. |
| (4) | Just the process of Scrambling is used whereas no text is hidden inside. | This secures the image to a great extent. | Text is not hidden in this paper and only image security is being processed whereas this paper focuses on image security as well. |
| (5) | Steganography is further discussed with the optimization techniques. | The optimization techniques are introduced so that the data encryption is more successful but the image remains as it is. | Since the image remains the same, the image is not secured enough which is one of the main points discussed in the paper above. |
| (6) | The author talks about hiding image inside another image to secure that. | Here, the concept varies between data security and image security itself. An image is secured inside another image. | Data security is not discussed here and image security is the most discussed here. |

*Continued on next page*

*Table 2 continued*

| | | | |
|---|---|---|---|
| [7] | This paper is about hiding data in image and also keeping the image clarity the same. | The clarity of the output image here is more when compared to other steganography methods. | By maintaining the clarity of the image, the possibility of the image being misused is more and hence, we need to provide security to the image as well. |
| [8] | Quantum Steganography is used to provide security to data. | Quantum steganography is discussed here with slight deviation from the concept of actual data security. | The topic leads to an all-new topic and is a tedious process. Also, the concept of image security is not discussed here. |
| [9] | Here, it is about storing data in a block of image and then securing them. | The data is stored in a block of image inside the whole image. This way the data remains more secure since the block needs to be found to decode the data in it. | The text is stored in a block of image instead of the whole image. This means that if that particular block is found, the data could be revealed. This method can be considered more because the possibility of finding the particular block remains low. |
| [10] | This paper talks about the use of Steganography and then Scrambling them, where the scrambled image is then used in steganography using graph wavelet transform method. | The graph wavelet transform method is used and the scrambled image is sent to steganography. | Even if the steganography process is undone, the image is scrambled. Which means that only unscrambling the image would not give us data. This differs from the discussion above as the process is reversed. |
| [11] | It states the different ways through which Steganography is optimized. Either done to receive embedding efficiency or for the distortion function that relates to the statistical detections in the image. | They recommend that authors try to improve the embedding efficiency rather than focusing on the distortion function. | In this paper, for the Steganography process, the data embedding is considered at first followed by the clarity of the image. And the clarity of the image is maintained throughout the process. |

As mentioned in Table 2, for [3], the resultant PSNR values have been mentioned as high, whereas in this paper, the resultant PSNR values are within the acceptable values threshold.

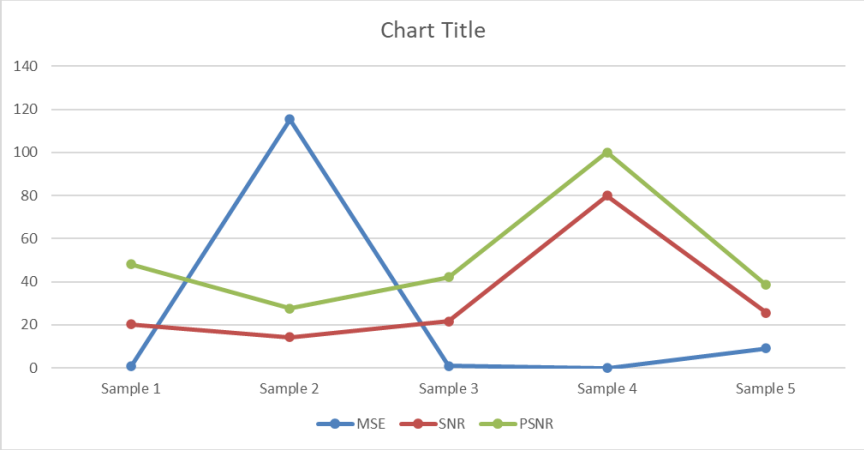
Here, 5 out of the 40 sample's PSNR, MSE values are provided in Table 3 for comparison and further discussion.

**Table 3. MSE, SNR and PSNR values for 5 out of the 40 samples used for the research**

| Sample-> | Sample -1 | Sample -2 | Sample -3 | Sample -4 | Sample -5 |
|---|---|---|---|---|---|
| **MSE** | 1.0 | 115.33333333333333 | 1.0 | 0.0 | 9.0 |
| **SNR** | 20.15623 dB | 14.23654 dB | 21.56463 dB | 80 dB | 25.48764 dB |
| **PSNR** | 48.1308036086791 dB | 27.511255167947965 dB | 42.18030061 dB | 100 dB | 38.58837851428585 dB |

Here, when the MSE is 0.0, it relays that there is no noise present in the processed image. Hence, PSNR could turn up to infinity, but it doesn't have any importance. Likewise, when PSNR stays between 20dB to 30dB, it is considered to be valid for wireless transmission. Above 30dB up to 50dB are also acceptable values here. The same is plotted in the Graph 1 shown below. For an image, SNR is considered acceptable between 20 dB – 40 dB for normal images. For images with very high quality, the SNR can result even from 50 dB to 100 dB.

As seen in Graph 1, the peak value in PSNR, SNR is obtained when MSE is low and the lowest value is obtained when MSE is high. And through the images shown as output, the blur is present for images with more PSNR and SNR. This gives us image security after Scrambling is done on the Steganography output. Hence, the objective of hiding data in image and receiving the image output with same clarity is achieved through Image Steganography and to blur or reduce the viewability of the image once Image Scrambling is done is achieved successfully.




Graph 1: **Plotting and Comparison of the MSE, SNR and PSNR values of the given 5 samples**

## 4 Conclusion

In the proposed hybrid technique, Steganography and Scrambling have been combined by embedding land ownership data in the document image which gives more data security and scrambling the image to provide with image security. The resultant images give more blur, thus providing more security to the data. For the samples specified here, the PSNR values ranges from 28 dB - 100 dB, whereas the acceptable values for PSNR is between 30 dB – 50 dB . Likewise, the acceptable range for SNR is from 20 dB – 40 dB and the results here are ranged from 14 dB -80 dB. The values for PSNR from 50 dB – 100 dB and SNR from 50 dB -100 dB would mean that the image doesn't have any noise and is of more clarity. The MSE values are the basic ones where 0 would be lowest and it would mean that the image doesn't have any distortion and is clear.

There is a question that would need to be addressed in the future regarding the security of the text data before being embedded into the image. Also, the amount of blur in the images needs to be regulated so that the resultant image doesn't get more distorted in the process. These processes can be discussed and concentrated on in the future for the research to be more optimal and successful. Likewise, maintaining the PSNR values between 30 dB – 50 dB and SNR values between 20 dB – 40 dB would be more reasonable and acceptable.

## References

1) Mandal PC, Mukherjee I, Paul G, Chatterji BN. Digital image steganography: A literature survey. *Information Sciences*. 2022;609:1451–1488. Available from: https://dx.doi.org/10.1016/j.ins.2022.07.120.
2) Subramanian N, Elharrouss O, Al-Maadeed S, Bouridane A. Image Steganography: A Review of the Recent Advances. *IEEE Access*. 2021;9:23409–23423. Available from: https://dx.doi.org/10.1109/access.2021.3053998.
3) Kaur S, Bansal S, Bansal RK. Image steganography for securing secret data using hybrid hiding model. *Multimedia Tools and Applications*. 2021;80:7749–7769. Available from: https://dx.doi.org/10.1007/s11042-020-09939-7.
4) Hosny KM, Kamal ST, Darwish MM. A color image encryption technique using block scrambling and chaos. *Multimedia Tools and Applications*. 2022;81:505–525. Available from: https://dx.doi.org/10.1007/s11042-021-11384-z.
5) Gnanalakshmi V, Indumathi G. A review on image steganographic techniques based on optimization algorithms for secret communication. *Multimedia Tools and Applications*. 2023;82:44245–44258. Available from: https://dx.doi.org/10.1007/s11042-023-15568-7.
6) Huo L, Chen R, Wei J, Huang L. A High-Capacity and High-Security Image Steganography Network Based on Chaotic Mapping and Generative Adversarial Networks. *Applied Sciences*. 2024;14(3):1–19. Available from: https://dx.doi.org/10.3390/app14031225.
7) Chen Z, Ma C, Feng Y, Hou X, Qian X. Directional lifting wavelet transform domain image steganography with deep-based compressive sensing. *Multimedia Tools and Applications*. 2023;82(26):40891–40912. Available from: https://dx.doi.org/10.1007/s11042-023-14939-4.
8) Yao JL, Yang HM, Jiang DH, Yan B, Pan JS, Wang MX. A Novel Quantum Image Steganography Algorithm Based on Double-Layer Gray Code. *International Journal of Theoretical Physics*. 2023;62(3). Available from: https://dx.doi.org/10.1007/s10773-023-05303-1.
9) Khandelwal J, Sharma VK. W-VDSR: wavelet-based secure image transmission using machine learning VDSR neural network. *Multimedia Tools and Applications*. 2023;82(27):42147–42172. Available from: https://dx.doi.org/10.1007/s11042-023-15166-7.
10) Sharma VK, Sharma PC, Goud H, Singh A. Hilbert quantum image scrambling and graph signal processing-based image steganography. *Multimedia Tools and Applications*. 2022;81(13):17817–17830. Available from: https://dx.doi.org/10.1007/s11042-022-12426-w.
11) Abdulla AA. Digital image steganography: challenges, investigation, and recommendation for the future direction. *Soft Computing*. 2023;p. 1–14. Available from: https://dx.doi.org/10.1007/s00500-023-09130-8.