*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 4*
*DOI: https://doi.org/10.17762/ijritcc.v11i4.6394*
***Article Received**: 03 February 2023 **Revised**: 05 March 2023 **Accepted**: 12 March 2023*

# Encryption and Decryption of Images with Pixel Data Modification Using Hand Gesture Passcodes

**B Pawan Kumar[1], N Harshavardhan Reddy[2], Rashmi Ranjan Das[3], Debopam Dey[4], Deepika Rani Sona[5], Aarthy M[6]**
[1]School of Electronics Engineering
Vellore Institute of Technology
Vellore,India
bollinenipaone@gmail.com
[2]School of Electronics Engineering
Vellore Institute of Technology
Vellore,India
harshahvr999@gmail.com
[3]School of Electrical Engineering
Vellore Institute of Technology
Vellore,India
rashmiranjandas@vit.ac.in
[4]Moonwalkwers Funlabs Private Ltd.
Bengaluru, India
debopam@zoomies.in
[5]School of Electronics Engineering
Vellore Institute of Technology
Vellore,India
deepika.rs@vit.ac.in
[6]School of Electronics Engineering
Vellore Institute of Technology
Vellore,India
aarthy.m@vit.ac.in

**Abstract**— To ensure data security and safeguard sensitive information in society, image encryption and decryption as well as pixel data modifications, are essential. To avoid misuse and preserve trust in our digital environment, it is crucial to use these technologies responsibly and ethically. So, to overcome some of the issues, the authors designed a way to modify pixel data that would hold the hidden information. The objective of this work is to change the pixel values in a way that can be used to store information about black and white image pixel data. Prior to encryption and decryption, by using Python we were able to construct a passcode with hand gestures in the air, then encrypt it without any data loss. It concentrates on keeping track of simply two pixel values. Thus, pixel values are slightly changed to ensure the masked image is not misleading. Considering that the RGB values are at their border values of 254, 255 the test cases of masking overcome issues with the corner values susceptibility.

**Keywords**—Pixel, Masking, Cypher text,Stego Image, Passcode.

## I. INTRODUCTION

With the increase in demand for data security, there is a need for new algorithms and techniques to mask data, so that even if data gets leaked the user would not be able to decode the information to use it. This paper mainly focuses on pixel color values of the image, and how it can be incremented and decremented without changing the color of the whole pixel. Secret data can be inserted into an image using the image hiding technique without affecting the original picture. A stego-image is an image that hides secret information. The stego-image won't raise any alerts, enabling the prevention of attacks. But the secret information concealed in the stego-image can be effectively decoded by the appropriate recipient . As we know most of the colors can be formed by mixing red, green and blue, deriving the pixel values in terms of RGB values, and changing respective intensities by a small desired amount can help us store information.

This is how we have designed a program that would be able to extract the image features, that is the pixel color with respect to the RGB values and we would be changing these RGB values according to the image that needed to be stored[1]. Instead of using the standard least significant bit(LSB) substitution techniques[2], we would be directly modifying the intensity of the color. This paper proposes a simple method to store multiple black-and-white images in a single color image. In Section 2, the existing techniques are reviewed. Additionally, Section 3 presents the proposed method's design. The effectiveness of the masking method is demonstrated in Section 4 along with the results of tests on numerous examples. The conclusion is presented in Section 5.

## II. RELATED WORKS

There are several ways to hide an image, including spatial domain data hiding, in which we directly deal with changing the image's pixel values[3]–[5]. Compressed domain hiding where both the compression of the images are done and this act as the encoding layer which is required to decode the masked image. Frequency domain hiding where the image is coded into frequency domain on which the image masking is done and the result is obtained in the frequency domain on which inverse transformation is applied to get back the encoded images[6]. Section 5 provides the conclusion of the work. Also with respect to data hiding various wavelet transform techniques came into play where OFDM systems along with, wavelet transform which is used for encoding and decoding are used [7]–[9] . The authors have put forth a neural network-based digital watermarking method that hides images. The purpose of neural networks is to make the image least perceptible to human eyes. Wavelet transform techniques have also been used to hide audio signals inside the image where the audio signals are converted to wav format and then encoded using wavelet transform[10]. There were also attempts where multibit assignment image data hiding for palette images have been made, where colors of the palette were used to hide secret data bits which hold information. Graph theories such as depth-first search were used in the process [11].

When accomplishing data hiding, some factors that must be considered includes the method's ability to hide data, which should be applicable to any sort of image, strong visual quality, low MSR and highpeak signal to noise ratio (PSNR) values, and stenographic security. Applications, where there is in need of high data capacity the proposed methods having high data hiding embedding capacity have been used where the visual quality is also been maintained or image quality is traded for more hiding capacity. Some papers have taken robustness into account where robust image data hiding schemes which use a statistical quantity to reduce the effects of image compression have been developed [12] . The other parameter taken into account for some proposed methods is the image visual quality which can be measured in terms of PSNR values of the masked images where the algorithm chooses and modifies the minimum points of the histogram of the image by a small amount to hide data [9]. Many attempts were made to prove stenography security, authors proposed a LSB matching in steganography in grayscale images where the correlation between pixels of natural images have been measured and made sure this is not altered before and after data hiding using various techniques[10]. Authors have proposed ideas where improved security measures have been implemented between user to user communication, secret image sharing schemes(SIS) identity based authentication have been proposed where the division of image into shadows have been implemented where the count of shadows is with regard to user identification number for improved security [13],[14].

Many authors have proposed methods to store grayscale pixels by encoding each pixel into 8 bits, then stored inside the hidden image. The key drawback of this approach is that it would store much less data than more recent encryption techniques [15], [16]. A hybrid data hiding scheme that combines the LSB technique with the key permutation method and a new method for determining the best key-permutation by using gene expression programming (GEP) has been proposed in the literature. However, one drawback is that the amount of data stored is less when compared to the resolution of the image [17] .Some proposed methods were used to store medical data such as X-rays, ECG data, personal details, prescriptions etc and this was used to make sure of medical security. This uses simple LSB substitution techniques, data is stored as bit per bit so the amount of data stored is less [18]. Moreover, solutions utilising an adaptive data-hiding technique in the frequency domain were put forth. They used the Haar Discrete Wavelet Transform (HDWT) to breakdown each block into a number of bands after dividing the original image into 88 sub-blocks.The amount of data stored is significantly more than other proposed methods but the processing time is more [19].

The majority of the publications suggested have created LSB-based methods. Several of them first used enhanced edge detection filters to detect edges in the cover image while encrypting the hidden data. The least significant byte of randomly chosen edge area pixels and 1-3-4 LSBs of the red, green, and blue components, respectively, of randomly chosen pixels throughout smooth portions of the image are then used to embed message bits [20]. For an image during the encryption stage, several researchers developed an enhanced reversible data hiding (RHD) technique [21]. To increase data security, the author used a stream cypher with a block permutation during the encryption process. A built-in embedding flag, error-free decryption, and a high embedding rate are further benefits of this research. Thus, the recipient needs both the image encryption key and the data-hiding key [22]. In these types of data-hiding techniques, the data is hidden and modified at the binary level. Whereas in our proposed method, we have encrypted the image using the RGB concentrations. So, this process can be used to store 3 completely different kinds of black-and-white images in one image. The recipient must have both the masked and unmasked images to retrieve the hidden information. Some authors have proposed a hybrid data hiding scheme that uses LSB and permutation methods and also used gene expression programming to store text data in images [23].

Our work supports the representation of data as images. Some articles offered dual image hiding in the cover image, where various techniques are utilised for individual images, including lifting wavelet transform (LWT) and discrete wavelet transform (DCT) [24]. There were also papers which proposed image hiding based on image histogram statistics where cumulative peak values of histograms are used for storing data and parameters like PSNR values were also taken into care to improve visual aspects of the image [25] .The authors proposed ANN artificial neural network architecture(ANN), which is used in conjunction with MPEG-7 texture decoder to retrieve the masked data from the cover image using texture as the comparison parameter. We also need effective decoding methods in case the user is unable to obtain the decryption key [26]. In order to obtain maximum efficiency in terms of embedding rate, visual quality and more some authors have used multiple methods like reversible hiding, histogram shifting, interpolation and difference expansion for data hiding in images [27], [28].

## III. METHODOLOGY

The proposed method requires both the masked image and the unmasked image. The pixel data of both images must be compared to retrieve the hidden data. The steps involve are:

### A. *Creating a passcode with hand gestures*

First of all, we create our own required passcode with the movement of our index finger in the air and this could be done with the help of python coding. A small dot cursor is placed on the index finger and it follows the path that our finger moves and hence by creating a design of the passcode. This is shown in Fig. 1.

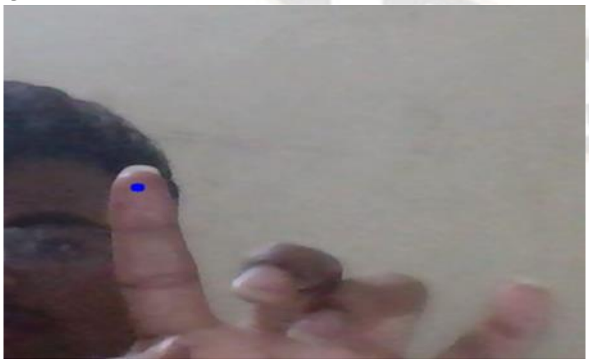


Fig.1 Illustrates the movement of the finger and creation of the passcode which is going to be masked with an image through the process of encryption.

### B. *Converting to black and white image*

If a coloured image is sent, it will be changed to a grayscale version. The three RGB colour characteristics that make up a coloured image classify it as a three-dimensional array. As a result, a three dimensional array is transformed into a two dimensional array since a grayscale image only has two colors—black and white—with varying intensities of each. This can be done using a basic equation(1):

$$Grayscale\ pixel\ value = \frac{R + G + B}{3} \qquad (1)$$

where R, G, and B are the hues of red, green, and blue, respectively. However, encoding 255 different intensities into the image would require change in pixel values by a higher value, so reducing this into only two intensities would reduce the change in pixel values required. This can be achieved using a threshold value ; which can be chosen as the center value of 0 to 255 , that is 128. Or else one can choose the threshold as mean value, which can help to properly encode image.

The final grayscale image would then be encoded as follows: if the grayscale value is less than the threshold value, the pixel value would be coded as "0," and if the value is more than the threshold, the pixel value would be coded as "255."

### C. *Encoding black and white image onto colored cover image*

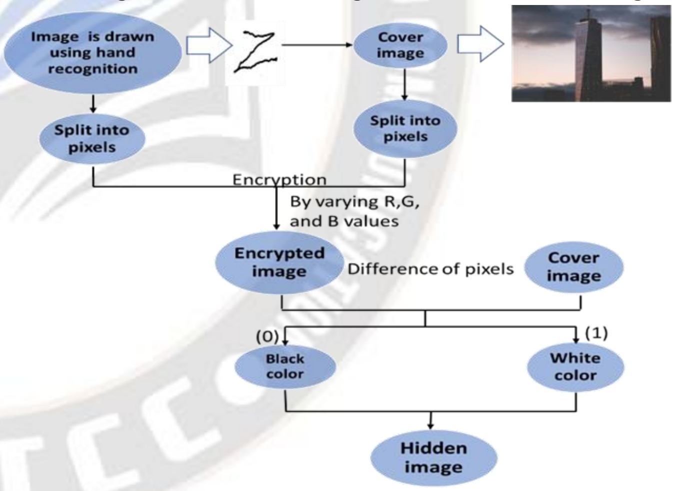


Fig. 2 System model for image encoding onto colored cover image.The cover image is divided into small pixels and encrypted with the image drawn using hand recognition and decrypted the image using difference of pixels from cover image and encrypted image.

Fig.2 depicts the flow of the proposed method. The final converted colour image consists of only two color values, that is 0 and 255 in a two dimensional array; these values can be treated as binary values. But the masking is not done with respect to LSB substitution. The pixel values that are the RGB intensities are modified to store these image values. Instead of storing these binary values in the least significant bit or kth-bit of the pixel data. The RGB values are incremented and decremented as follows:

If the pixel value is taken as '0' then the one color intensity value is selected , say Blue colour value is incremented by 1. But to reduce the outliers susceptibility that is if 'B' value is at the end value that is at 255, then the value must be decremented by the same amount '1'to reduce drastic change in pixel colour. Similarly if the pixel value is '255' then the same colour; here 'B' is selected and its value would be incremented by '2' , but in this case if the unchanged value of 'B' is 254 or 255 then the value must be decremented by '2'. This ambiguity at outliners occurs because, if the value is 255 and it is incremented by 1, this leads to 256 which is not possible in terms of pixel color value. So, it would be considered as '0', which would change the complete color of the image pixel.

### *D. Encoding multiple black and white images into single coloured image*

With respect to coloured images, we have three different colours for the pixel value. So, one can modify these individual colour intensity values. This can assure us a minimum of 3 different images in one coloured image. But if the cover image size is large enough then one can process different areas of the image to hide multiple images. Say if the size of cover image is (1000x1000) pixels, but the hidden images has sizes say (250x250) , (200x200), and (500x500) the remaining pixels data that is (750x750) ,(800x800) and (500x500) would be unused, so we can loop through these pixels and store at least 3 more images. So, the utilization of unused pixels can help to store multiple images.

### *E. Retrieving data from the masked (stego-image) image by comparing it with the original image:*

After sending the masked image to the intended user, one must have access to the original image. So, choosing an accessible original image also plays an important role. Retrieving data can be done by comparing the masked image with the original image. If only one image is stored with respect to a single colour in RGB , the user can compare each pixel value data, and get two different values. So, conversion of data to image can be done as follows:

• If the difference between the original and masked pixel colour values differ by '1', then one must consider the pixel colour as (0,0,0) and if the difference is '2' the pixel color as (255,255,255).

• But, if the number of images masked per color is more than 1,extracting data directly would have multiple images, so one must know the dimensions of the individual images to split the images. We have performed the test cases for the number of images stored is 3 because the size of hidden images chosen is equal to stego image.

## IV. RESULTS

The amount of data stored can be a minimum of 3 times the size of the cover image pixels. Here the data hidden are chosen as images so 3 images are stored. The respective results shown in this section where we have considered the commonly used test stego images for testing. Fig. 3 shows some stego images from the existing research [15].

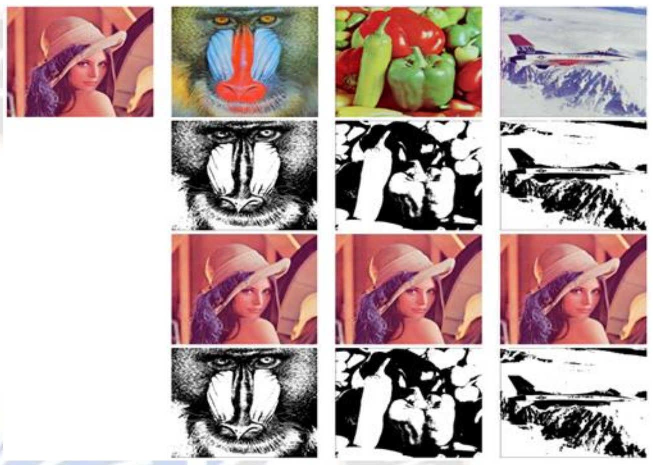


Fig. 3 Stego Images used for test cases.

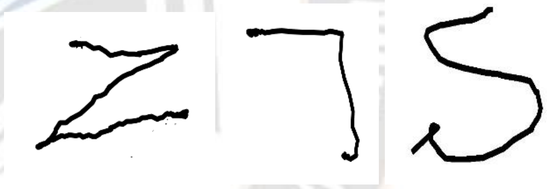


Fig. 4 Fig. 5 Fig. 6
Fig. 4,Fig. 5 and Fig. 6 represents the passcode that was created by us with our hand gestures and now this image is going to be encrypted with a coloured image.

Fig. 7 Final mask image for decryption.

Fig. 7 is the image which contains our secret image/pass codes and is used as a mask and can be decrypted to get our original image.

A. *Quantative analysis using PSNR and MSE values for hand gestures*

Table I and Table II shows the PSNR and mean square error (MSE) values of created passcode as hidden images.

TABLE I PSNR values of hidden images.

| **Cover image** | **No of images hidden** | **PSNR(db)** |
|---|---|---|
| Fig 7 | 1(Fig 4) | 47.10412438536813 |
| Fig 7 | 2(Fig 4+Fig 5) | 44.04958037328089 |
| Fig 7 | 3(Fig4+Fig5+Fig6) | 42.31854563909316 |

TABLE II MSE values of hidden images.

| **Cover image** | **No of images hidden** | **MSE** |
|---|---|---|
| Fig 7 | 1(Fig 4) | 1.2666829427083333 |
| Fig 7 | 2(Fig 4+Fig 5) | 2.559306640625 |
| Fig 7 | 3(Fig 4+Fig 5+Fig 6) | 3.8126399739583334 |

B. *Number of bits stored*

The following 4 tests are performed to calculate number of bits stored .

**Test1:**

Test Images:

- *Baboon- (black and white image)=32768 bytes*
- *Peppers- (black and white image)=32768 bytes*
- *Airplane- (black and white image)=32768 bytes*
- *Lena- Coloured image(Stego Image)*

**Test2:**

Test Images:

- *Lena- (black and white image)=32768 bytes*
- *Peppers- (black and white image)=32768 bytes*
- *Airplane- (black and white image)=32768 bytes*
- *Baboon- Coloured image(Stego Image)*

**Test3:**

Test Images:

- *Baboon- (black and white image)=32768 bytes*
- *Lena- (black and white image)=32768 bytes*
- *Airplane- (black and white image)=32768 bytes*
- *Peppers- Coloured image(Stego Image)*

**Test4:**

Test images

- *Baboon- (black and white image)=32768 bytes*
- *Peppers- (black and white image)=32768 bytes*
- *Lena- (black and white image)=32768 bytes*
- *Airplane- Coloured image(Stego Image)*

Total number of bits stored in each test performed=32768*3=98304 bytes.

C. *Quantitative analysis using PSNR and MSE values for images*

The maximum number of Images that can be hidden which have equal size to the cover image is 3.Table III shows the PSNR and MSE values when Lena is used as a stego image for test.

TABLE III When the stego image is Lena

| **No of images hidden** | **PSNR(db)** | **MSE** |
|---|---|---|
| 1 | 47.980531136122195 | 1.0352071126302083 |
| 2 | 45.33118120776055 | 1.9052950541178386 |
| 3 | 43.75258190892207 | 2.7404518127441406 |

Table IV shows the PSNR and MSE values when Baboon is used as a stego image for test.

TABLE IV When the stego image is Baboon

| **No of images hidden** | **PSNR(db)** | **MSE** |
|---|---|---|
| 1 | 47.980531136122195 | 1.0352071126302083 |
| 2 | 45.3972168280825 | 1.876543680826823 |

| | | |
|---|---|---|
| 3 | 43.742799450309484 | 2.746631622314453 |

Table V shows the PSNR and MSE values when Peppers image is used as a stego image for test.

TABLE V When the stego image is Peppers

| No of images hidden | PSNR(db) | MSE |
|---|---|---|
| 1 | 48.91312360813345 | 0.8351567586263021 |
| 2 | 45.812936524461946 | 1.7052447001139324 |
| 3 | 44.071228206205284 | 2.546581268310547 |

Table VI shows the PSNR and MSE values when airplane image is used as a stego image for test.

TABLE VI When the stego image is an airplane

| No of images hidden | PSNR(db) | MSE |
|---|---|---|
| 1 | 47.980531136122195 | 1.0352071126302083 |
| 2 | 45.3972168280825 | 1.876543680826823 |
| 3 | 43.798386493742676 | 2.711700439453125 |

From Fig. 8, we can see a difference in PSNR from test1,test2, and test3 with PSNR from test4 the reason behind it is the cover image that we have used in test 4 has the major portion of white color in the image. The RGB concentration for white(255,255,255) which is higher compare to cover images colors of test1,test2,test3.

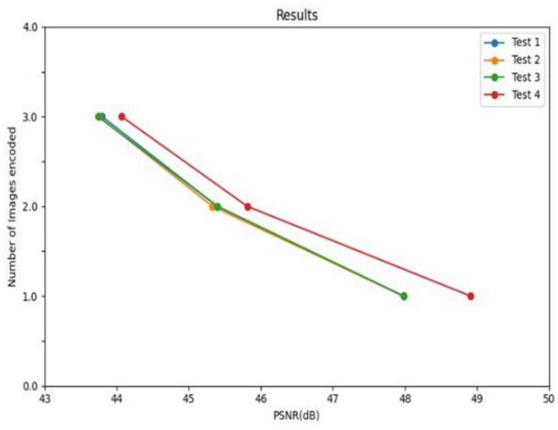
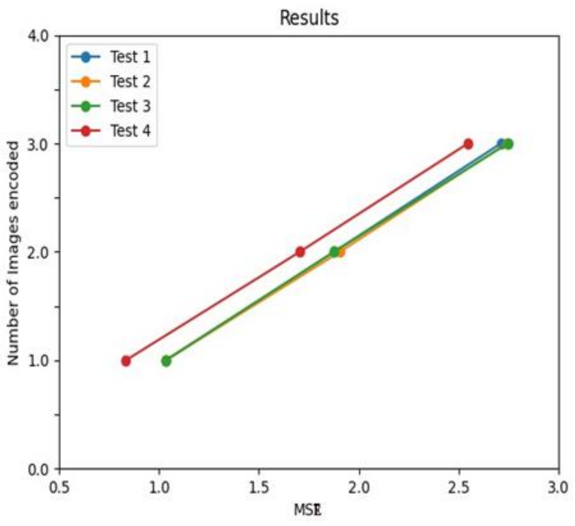


Fig. 8 PSNR Vs. The number of images hidden for tests performed.



Prepare Your Fig. 9 MSE Vs.Number of images hidden for tests performed.

From Fig. 9 we can observe low MSE for test4 compared to different tests the reason behind it is a color image used in test 4 has more RGB concentration compare to the color image used in the remaining tests.

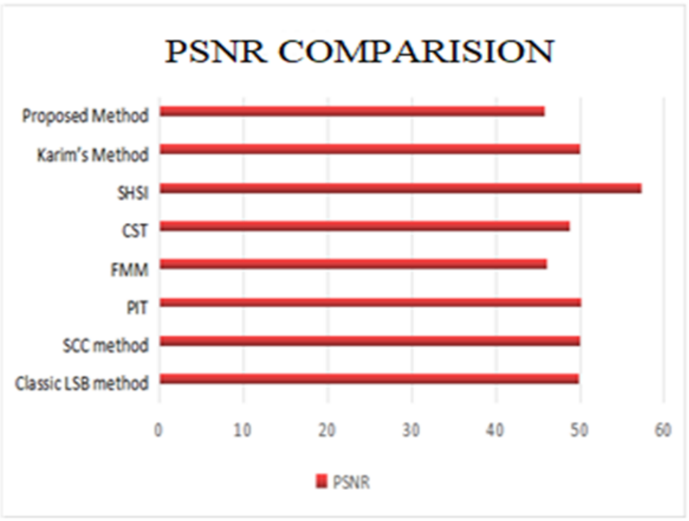


Fig. 10 PSNR Comparison of existing method to Proposed method.

Fig. 10, illustrates our suggested approach stores three images in a single image, but the other methods can only store and retrieve one image. Despite this, our proposed methodology has relatively comparable PSNR to the other approaches. Loss of image quality, compatibility, and complexity are some other frequent major problems that are resolved in our method, and we also addressed the restricted capacity issue, which is the main drawback in all other approaches where only 1 image can be stored.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 4*
*DOI: https://doi.org/10.17762/ijritcc.v11i4.6394*
***Article Received**: 03 February 2023 **Revised**: 05 March 2023 **Accepted**: 12 March 2023*

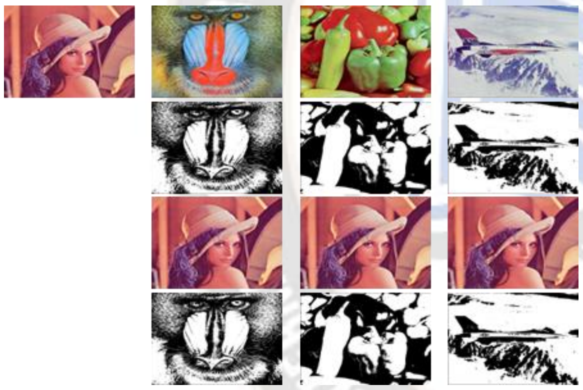
### D. Test Results Obtained after application of methodology

**TABLE VII** Lookup table for below test results

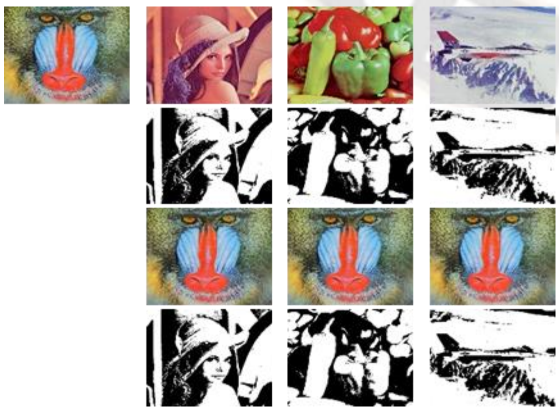
| | **Hidden image 1** | **Hidden image 2** | **Hidden image 3** |
|---|---|---|---|
| **Stego Image** | Black and white Converted hidden image 1 | Black and white Converted hidden image 2 | Black and white Converted hidden image 3 |
| | Stego-image after masking image 1 | Stego-image after masking image1,2 | Stego-image after masking image 1,2,3 |
| | Retrieved image 1 | Retrieved image 2 | Retrieved image 3 |

The look-up table in Table VII provides a brief explanation of the cases where we used a different image as the stego-image in each case and masked three images. The table outlines the procedures, such as changing each image to black and white, hiding them one at a time, and then retrieving them.

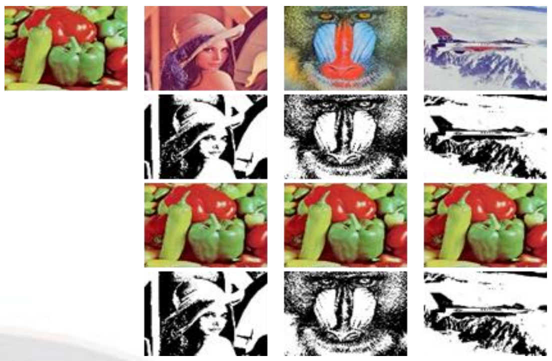
TEST 1



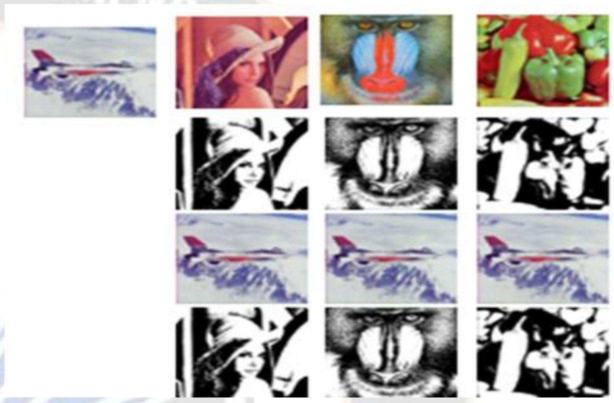
TEST 2



TEST 3



TEST 4



## V. CONCLUSION

In this research, we present a simplified data hiding method for storing black and white that we have provided using hand movements and stored. These Black and white images were stored inside the color images with high PSNR ratio with respect to ".png" images.This method consists of modifying the stego image pixel data that is modifying the RGB concentrations of each pixel which would hold the hidden information. The problem of drastic change of pixel color has also been tackled. The proposed method can also be used to hold more than 3 images if the stego image size is large enough with proper segmentation.

## REFERENCES

[1] D. Lerch-Hostalot and D. Megías, "LSB matching steganalysis based on patterns of pixel differences and random embedding," Comput Secur, vol. 32, 2013, doi: 10.1016/j.cose.2012.11.005.

[2] M. N. Wu, M. H. Lin, and C. C. Chang, "A LSB substitution oriented image hiding strategy using genetic algorithms," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3309, 2004, doi: 10.1007/978-3-540-30483-8_27.

[3] S. Badr and P. Noufal, "Integrated Data Hiding and Compression Scheme Based on SMVQ and FoE Inpainting," Procedia Technology, vol. 24, 2016, doi: 10.1016/j.protcy.2016.05.220.

*International Journal on Recent and Innovation Trends in Computing and Communication*
***ISSN: 2321-8169 Volume: 11 Issue: 4***
***DOI: https://doi.org/10.17762/ijritcc.v11i4.6394***
***Article Received**: 03 February 2023 **Revised**: 05 March 2023 **Accepted**: 12 March 2023*

[4] C. C. Chang, T. D. Kieu, and W. C. Wu, “A lossless data embedding technique by joint neighboring coding,” Pattern Recognit, vol. 42, no. 7, 2009, doi: 10.1016/j.patcog.2008.11.040.

[5] C. C. Chang, T. D. Kieu, and Y. C. Chou, “Reversible information hiding for VQ indices based on locally adaptive coding,” J Vis Commun Image Represent, vol. 20, no. 1, 2009, doi: 10.1016/j.jvcir.2008.08.005.

[6] S. Lee, C. D. Yoo, and T. Kalker, “Reversible image watermarking based on integer-to-integer wavelet transform,” IEEE Transactions on Information Forensics and Security, vol. 2, no. 3, 2007, doi: 10.1109/TIFS.2007.905146.

[7] S. I. Nipanikar and V. H. Deepthi, “Wavelet transform-based steganographic method for secure data communication using OFDM system,” International Journal of Intelligent Computing and Cybernetics, vol. 10, no. 3, 2017, doi: 10.1108/IJICC-11-2016-0051.

[8] S. I. Nipanikar, V. Hima Deepthi, and N. Kulkarni, “A sparse representation based image steganography using Particle Swarm Optimization and wavelet transform,” Alexandria Engineering Journal, vol. 57, no. 4, 2018, doi: 10.1016/j.aej.2017.09.005.

[9] T. Rabie, M. Baziyad, and I. Kamel, “Enhanced high capacity image steganography using discrete wavelet transform and the Laplacian pyramid,” Multimed Tools Appl, vol. 77, no. 18, 2018, doi: 10.1007/s11042-018-5713-2.

[10] S. Hemalatha, U. D. Acharya, and A. Renuka, “Wavelet transform based steganography technique to hide audio signals in image,” in Procedia Computer Science, 2015. doi: 10.1016/j.procs.2015.03.207.

[11] H. Wu, H. Wang, H. Zhao, and X. Yu, “Multi-layer assignment steganography using graph-theoretic approach,” Multimed Tools Appl, vol. 74, no. 18, 2015, doi: 10.1007/s11042-014-2050-y.

[12] S. N. Mali, P. M. Patil, and R. M. Jalnekar, “Robust and secured image-adaptive data hiding,” Digital Signal Processing: A Review Journal, vol. 22, no. 2, 2012, doi: 10.1016/j.dsp.2011.09.003.

[13] P. Wang, X. He, Y. Zhang, W. Wen, and M. Li, “A robust and secure image sharing scheme with personal identity information embedded,” Comput Secur, vol. 85, 2019, doi: 10.1016/j.cose.2019.04.010.

[14] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, “Robust lossless image data hiding designed for semi-fragile image authentication,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 4, 2008, doi: 10.1109/TCSVT.2008.918761.

[15] Arun K Singh, “Steganography in Digital Images Using LSB Technique,” Journal of Xidian University, vol. 14, no. 5, pp. 1–5, 2020.

[16] G. Swain, “A Steganographic Method Combining LSB Substitution and PVD in a Block,” in Procedia Computer Science, 2016. doi: 10.1016/j.procs.2016.05.174.

[17] S. Gapper et al., “Data Hiding by LSB Substitution Using Genetic Optimal Key-Permutation.,” Inf Sci (N Y), vol. 2, no. 3, 2016.

[18] R. Karakiş, I. Güler, I. Çapraz, and E. Bilir, “A novel fuzzy logic-based image steganography method to ensure medical data security,” Comput Biol Med, vol. 67, 2015, doi: 10.1016/j.compbiomed.2015.10.011.

[19] B. L. Lai and L. W. Chang, “Adaptive data hiding for images based on Harr Discrete Wavelet Transform,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2006. doi: 10.1007/11949534_109.

[20] M. Juneja and P. S. Sandhu, “An Improved LSB Based Steganography Technique for RGB Color Images,” International Journal of Computer and Communication Engineering, 2013, doi: 10.7763/ijcce.2013.v2.238.

[21] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, 2006, doi: 10.1109/TCSVT.2006.869964.

[22] Q. Li, B. Yan, H. Li, and N. Chen, “Separable reversible data hiding in encrypted images with improved security and capacity,” Multimed Tools Appl, vol. 77, no. 23, 2018, doi: 10.1007/s11042-018-6187-y.

[23] C. K. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” Pattern Recognit, vol. 37, no. 3, 2004, doi: 10.1016/j.patcog.2003.08.007.

[24] K. Upendra Raju and N. Amutha Prabha, “Dual images in reversible data hiding with adaptive color space variation using wavelet transforms,” International Journal of Intelligent Unmanned Systems, vol. 11, no. 1, 2023, doi: 10.1108/IJIUS-08-2021-0095.

[25] F. A. Al-Omari, O. D. Al-Khaleel, G. A. Rayyashi, and S. H. Ghwanmeh, “An innovative information hiding technique utilizing cumulative peak histogram regions,” Journal of Systems and Information Technology, vol. 14, no. 3, 2012, doi: 10.1108/13287261211255356.

[26] M. A. Pratt, S. Konda, and C. H. Henry Chu, “Texture-based image steganalysis by artificial neural networks,” International Journal of Intelligent Computing and Cybernetics, vol. 1, no. 4, 2008, doi: 10.1108/17563780810919122.

[27] T. C. Lu, C. C. Chang, and Y. H. Huang, “High capacity reversible hiding scheme based on interpolation, difference expansion, and histogram shifting,” Multimed Tools Appl, vol. 72, no. 1, 2014, doi: 10.1007/s11042-013-1369-0.

[28] S. Y. Shen and L. H. Huang, “A data hiding scheme using pixel value differencing and improving exploiting modification directions,” Comput Secur, vol. 48, 2015, doi: 10.1016/j.cose.2014.07.008.