iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** II **Month of publication:** February 2017

DOI: http://doi.org/10.22214/ijraset.2017.2034

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 5 Issue II, February 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Edge Analysis Based Data Hiding on Digital Images with PVD Technique

Akhil P V[1], Dr Shelbi Joseph[2], Seetha Parameswaran[3]
[1,2,3]*Division of Information Technology, SOE, CUSAT*

Abstract*: *This paper proposes a new improved steganographic technique based on pixel value differencing[14] which is driven by edge recognition technique. The cover image is divided into blocks and rotated first to generate edges[10] in the image and is used as the candidate region to perform data hiding using pixel value differencing. The basic idea of pixel value differencing is a cover image is partitioned into non-overlapping blocks of two consecutive pixels. A difference value is calculated from the values of the two pixels in each block. All possible difference values are classified into a number of ranges. The difference value then is replaced by a new value to embed the value of a sub-stream of the secret message. The number of bits which can be embedded in a pixel pair is decided by the width of the range that the difference value belongs to. The method is designed in such a way that the modification is never out of the range interval. This method provides an easy way to produce a more imperceptible result than those yielded by simple least-significant-bit replacement methods. The embedded secret message can be extracted from the resulting stego-image without referencing the original cover image. Moreover, a pseudorandom mechanism may be used to achieve secrecy protection.

Keywords*: *Image steganography, LSB encoding, Pixel Value Differencing, cover image, stego image, security

I. INTRODUCTION

In recent years, the security and the confidentiality of the sensitive data has become of prime and supreme importance due to the explosive growth of internet and the fast communication techniques. Therefore how to protect secret messages during transmission becomes an important issue and hiding data provides a good layer of protection on the secret message. One of the widely accepted data hiding technique is image steganography. Image steganography uses a digital image as cover media and hence it is called cover image. The data is hidden in the cover image and the resulting image is called stego image. The data can be extracted out from the stego image and the existence of a hidden message in the cover image is invisible. The embedding of data in an image can cause distortion in the cover image and this distortion caused by data embedding is called embedding distortion. A good data-hiding method should be immune to statistical and visual detection while providing an adjustable payload [1], [2]. There are a number of techniques available which can perform image steganography in a digital image and this paper focuses on analyzing the different techniques and proposing a method which can offer better results over the methods that are studied.

A. Digital Steganography

A digital steganographic encoder is shown on Figure 1. The message is the data that the sender wishes to keep confidential and can be text, images, audio, video, or any other data that can be represented by a stream of bits. The cover or host is the medium in which the message is embedded and serves to hide the presence of the message. This is also referred to as the message wrapper. The message embedding technique is strongly dependent on the structure of the cover. It is not required that the cover and the message have homogeneous structure.

The image with the secretly embedded message produced by the encoder is the stego-image. The stego-image should resemble the cover image under casual inspection and analysis. In addition, the encoder usually employs a stego-key(optional) which ensures that only recipients who know the corresponding decoding key will be able to extract the message from a stego-image.[3]

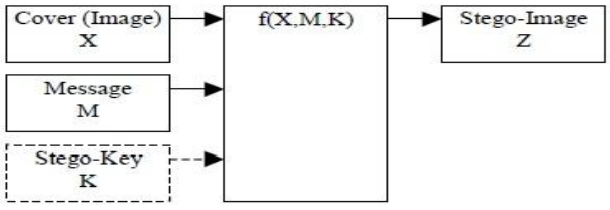


Fig 1 : Steganographic Encoding

B. Human Visual System and Image Steganography

The property of human eye on how an image is perceived is exploited by the human visual system (HVS). Certain characteristics of an image and not easily visible to human eye and those characteristics can be used and combined with image steganography to avoid visual and statistical detection of the embedded data. The key factors which influence them include:

The eye is less sensitive to noise in the high resolution bands and in those bands having orientation of 45.

The eye is less sensitive to noise in those areas of the image where brightness is high or low.

The eye is less sensitive to noise in highly textured areas, but, among these, more sensitive near the edges.

These properties can prove handy when combined and exploited in image steganography. Many of the data embedding techniques developed so far has concentrated only on the data hiding aspect of image steganography and not on studying the image characteristics and hiding the data.

The rest of this paper is organized as follows. Section 2studies the various methods developed for image steganography, a proposed system is presented in section 3 and section 4 includes conclusion.

II. SURVEY

There are various data hiding techniques developed in recent years. J. Mielikainen developed a method[4] which is based on pixel pair matching and it uses a pair of pixels as the embedding unit. The LSB of first pixel carries one bit of information and a binary function of the two pixel values carry another bit of information.This method offers same payload (number of bits embedded in the cover image) as LSB matching with fewer changes to cover image. The MSE of LSB for 1 bpp is 0.5,while for LSBMR it is 0.375.

OPAP [5] is an enhancement of LSB substitution method and it is based on embedding error. It uses only one pixel as embedding unit. In this method, for a m-bit pixel, if message bits are embedded to the right-most r LSB's then other m-r bits are adjusted by a simple evaluation. These m-r bits are either replaced by the adjusted result or otherwise kept unmodified based on if the adjusted result offers smaller distortion.

An improvement of LSB matching revisited method is EMD [6] in which each (2n+1)-ary notational system is carried by n cover pixels and at most only one pixel is increased or decreased by 1. The secret message is converted into a sequence of digits in the notational system with an odd base. Then pseudo-randomly permute all cover pixels according to a secret key, and divide them into a series of pixel-groups, each containing n pixels. The method is very well able to provide better stego image quality under the same payload than traditional LSB.

Diamond Encoding [7] an extension of EMD method and it first partitions the cover image into non-overlapping blocks of two consecutive pixels and transforms the message to a series of K-ary digits. For each block a Diamond Characteristic Value (DCV) is calculated and one secret K-ary digit is concealed into DCV. The DCV is modified to secret digit and it is done by adjusting pixel values in a block. This method is capable of hiding more secret data while keeping the stego-image quality degradation imperceptible.

Method used in [8]is an enhancement of the EMD method and this method segments the cover image into pixel sections and each section is partitioned into the selective and descriptive groups. The EMD embedding procedure is then performed on each group by referencing a predefined selector and descriptor table. The method combines different pixel groups of the cover image to represent more embedding directions with less pixel changes than the EMD method. By selecting the appropriate combination of pixel groups, the embedding efficiency and the visual quality of the stego image is enhanced. It offers higher embedding efficiency than EMD method.

Optimized Bit Plane splicing algorithm [9] is implemented by M. Naseem et.al. where in the pixels are grouped based on their intensity and then the number of bits are to represent the hidden data are chosen.As the bits are grouped based on the intensity of the pixels, more number of darker intensity pixels can be used to represent the hidden data than just the LSB.

Edge Adaptive Image steganography[10]extends the LSB matching revisited steganography. The method first divides the cover pixels into blocks and rotates each block by a random degree based on the secret key. The rotation causes new edges to appear and among the edges a threshold value is used to get the most suitable areas where data is hidden. Now the blocks are rotated back to normal and thus the data gets embedded. The reverse of the method is employed to extract the data. The method offers better image quality and immunity to steganalysis than the LSBMR method.

Reversible data embedding using interpolation and reference pixel distribution introduced in [11] is a reversible data hiding method based on image interpolation and detection of smooth and complex regions in the cover image. A binary image that represents the locations of reference pixels is constructed according to the local image activity. In complex regions, more reference pixels are

chosen and fewer pixels are used for embedding and vice-versa for smooth regions. Pixels are interpolated according to the constructed binary image, and interpolation errors are then used to embed data through histogram shifting. It offers better prediction and a mechanism to add or remove reference pixels based on local image characteristics. The method achieves better PSNR for a range of embedding rates.

Pixel Pair Matching(PPM) [12]method uses the values of pixel pair as a reference coordinate, and search a coordinate in the neighborhood set of this pixel pair according to a given message digit. The searched digit conceals the digit and it replaces the pixel pair. It makes use of a more compact neighborhood set than used in Diamond encoding. The extraction process finds the replaced pixel pair to extract the message data. Exploiting Modification Direction (EMD) method has a maximum capacity of 1.161 bpp and Diamond Encoding (DE) extends the payload of EMD by embedding digits in a larger notational system. The proposed method offers lower distortion than DE by providing more compact neighborhood sets and allowing embedded digits in any notational system. Compared with the optimal pixel adjustment process (OPAP) method, the proposed method always has lower distortion for various payloads.

The proposed method in [13] is a hiding scheme by replacing the LSB of a cover according to the difference values between a pixel and its four touching neighbours. Although this method can embed most secret data along sharper edges and can achieve more visually imperceptible stegos, the security performance is poor. Since the method just modifies the LSB of image pixels when hiding data, it can be easily detected by existing steganalytic algorithms, such as the RS analysis.

III.PROPOSED SYSTEM

Hiding data in the LSBs of the pixels of a gray valued image is a common information hiding method that utilizes the characteristic of the human visions insensitivity to small changes in the image. This simple LSB embedding approach is easy for computation, and a large amount of data can be embedded without great quality loss. The more LSBs are used for embedding, the more distorted result will be produced. Not all pixels in an image can tolerate equal amounts of changes without causing notice to an observer. The largest number of LSBs whose gray values can be changed without producing a perceptible artefact in each pixel is different. Changes of the gray values of pixels in smooth areas in images are more easily noticed by human eyes. In the embedding method we propose, we simply divide the cover image into a number of non-overlapping blocks and each block is rotated to a random degree determined by the secret key. This gives rise to new edges in the image and now we divide the edges into a number of non-overlapping two-pixel blocks. Each block is categorized according to the difference of the gray values of the two pixels in the block. A small difference value indicates that the block is in a smooth area and a large one indicates that it is in an edged area. The pixels in edged areas may, as mentioned previously, tolerate larger changes of pixel values than those in the smooth areas. So, in the proposed method we embed more data in edged areas than in the smooth areas. And it is in this way that we keep the changes in the resulting stego-image unnoticeable.

A. Extraction of Edges

The cover image of size of m x n is first divided into non overlapping blocks of Bz x Bz pixels. For each small block, we rotate it by a random degree in the range of {0,90,180,270}, as determined by a secret key key_1. The resulting image is rearranged as a row vector V by raster scanning. And then the vector is divided into non overlapping embedding units with every two consecutive pixels (x_i,x_{i+1}), where $i = 1,3,....,mn-1$, assuming n is an even number. Two benefits can be obtained by the random rotation. First, it can prevent the detector from getting the correct embedding units without the rotation key key_1, and thus security is improved. Furthermore, both horizontal and vertical edges (pixel pairs) within the cover image can be used for data hiding. The data hiding will be performed on the newly generated edges which are formed as a result of the rotation operation called as the candidate region. In order to ensure minimum susceptibility to steganalysis, data hiding is not performed directly on the candidate region, but on the candidate region we use the pixel value differencing technique to find the best pixels that can be used for data hiding process.

B. Quantization of differences of gray values of two-pixel blocks.

The cover images used in the proposed method are 256 gray-valued ones. A difference value d is computed from every non-overlapping block of two consecutive pixels, say pi and pi+1, of a given candidate region. The way of partitioning the candidate region into two-pixel blocks runs through all the rows of each image in a zigzag manner. Assume that the gray values of pi and pi+1are gi and gi+1, respectively, then d is computed as gi+1-gi, which may be in the range from -255 to255. A block with d close to 0 is considered to be an extremely smooth block, whereas a block with d close to -255 or 255 is considered as a sharply edged block. By symmetry, we only consider the possible absolute values of d (0 through 255) and classify them into a number of

contiguous ranges, say Ri where i = 1; 2; . . . ; n. These ranges are assigned indices 1 though n. The lower and upper bound values of Ri are denoted by li and ui, respectively, where l1 is 0 and un is 255. The width of Ri is ui - li +1. In our proposed method, the width of each range is taken to be a power of 2.This restriction of widths facilitates embedding binary data. The selected range intervals are based on the human visual capability mentioned previously. The widths of the ranges which represent the difference values of smooth blocks are chosen to be smaller while those which represent the difference values of edged blocks are chosen to be larger. That is, we create ranges with smaller widths when d is close to 0 and ones with larger widths when d is far away from 0 for the purpose of yielding better imperceptible results. A difference value which falls in a range with index k is said to have index k. All the values in a certain range (i.e., all the values with an identical index) are considered as close enough. That is, if a difference value in a range is replaced by another in the same range, the change presumably cannot be easily noticed by human eyes. In the proposed method, we embed some bits of the secret message into a two-pixel block by replacing the difference value of the block with one with an identical index, i.e., we change a difference value in one range into any of the difference values in the same range. In other words, in the proposed data embedding process, we adjust the gray values in each two pixel pair by two new ones whose difference value causes changes unnoticeable to an observer of the stego-image. More details are described next.

C. Data embedding

We consider the secret message as a long bit stream. We want to embed every bit in the bit stream into the two-pixel blocks of the cover image. The number of bits which can be embedded in each block varies and is decided by the width of the range to which the difference value of the two pixels in the block belongs. Given a two-pixel block *B* with index *k* and gray value difference *d*, the number of bits, say *n*, which can be embedded in this block, is calculated by . Since the width of each range is selected to be a power of 2, the value of $n = \log_2(u_k - l_k + 1)$ is an integer. A sub-stream S with n bits is selected next from the secret message for embedding in *B*. A new difference *d'* then is computed by

$$d' = \begin{cases} l_k + b & \text{for } d \geqslant 0; \\ -(l_k + b) & \text{for } d < 0, \end{cases}$$

where b is the value of the sub-stream S. Because the value b is in the range from 0 to $u_k - l_k$, the value of d' is in the range from l_k to u_k. According to the previous discussions, if we replace d with d', the resulting changes are presumably unnoticeable to the observer. We then embed b by performing an inverse calculation from d0 described next to yield the new gray values (g'_i, g'_{i+1}) for the pixels in the corresponding two-pixel block (p'_i, p'_{i+1}) of the stego-image. The embedding process is finished when all the bits of the secret message are embedded.

The inverse calculation for computing (g'_i, g'_{i+1}) from the original gray values (g_i, g_{i+1}) of the pixel pair is based on a function $f((g_i, g_{i+1}), m)$ which is defined to be $f((g_i, g_{i+1}), m) = (g'_i, g'_{i+1})$

$$f((g_i, g_{i+1}), m) = (g'_i, g'_{i+1}) = \begin{cases} (g_i - \text{ceiling}_m, g_{i+1} + \text{floor}_m) \\ \quad \text{if } d \text{ is an odd number;} \\ (g_i - \text{floor}_m, g_{i+1} + \text{ceiling}_m) \\ \quad \text{if } d \text{ is an even number,} \end{cases} \quad (2)$$

The above equation satisfies the requirement that the difference between g'_i and g'_{i+1} is d'. It is noted that a distortion reduction policy has been employed in designing Eq. (2) for producing g'_i and g'_{i+1} from g_i and g_{i+1} so that the distortion caused by changing g_i and g_{i+1} is nearly equally distributed over the two pixels in the block. The effect is that the resulting gray value change in the block is less perceptible.

In the above inverse calculation, a smaller value of *d'* produces a smaller range interval between g'_i and g'_{i+1} while a larger *d'* produces a larger interval. So, (g_i, g_{i+1}) may produce invalid (g'_i, g'_{i+1}), i.e.,some of the calculations may cause the resulting g'_i or g'_{i+1} to fall off the boundaries of the range [0,255]. Although we may re-adjust the two new values into the valid range of [0, 255] by forcing a falling-off boundary value to be one of the boundary values of 0 and 255, and adjusting the other to a proper value to satisfy the difference *d'*, yet this might produce abnormal spots in contrast with the surrounding region in some cases. To solve this problem, we employ a checking process to detect such falling off-boundary cases, and abandon the pixel blocks which yield such cases for data embedding. The gray values of the abandoned blocks are left intact in the stego-image. This strategy helps us to

Volume 5 Issue II, February 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

distinguish easily blocks with embedded data from abandoned blocks in the process of recovering data from a stego-image, which will be discussed in the next section. It is noted that such abandoned pixel blocks are very few in real applications according to our experiments.

The proposed falling-off-boundary checking proceeds by producing a pair of $(\hat{g}_i, \hat{g}_{i+1})$ from the inverse calculation of the value of the function $f((g_i, g_{i+1}), u_k\text{-}d)$. Since u_k is the maximum value in the range from l_k to u_k, the resulting pair of$(\hat{g}_i, \hat{g}_{i+1})$ produced by the use of u_k will yield the maximum difference. That is, this maximum range interval $\hat{g}_{i+1}$- g_i covers all of the ranges yielded by the other $(\hat{g}_i, \hat{g}_{i+1})$ pairs. So, the falling-off-boundary checking for the block can proceed by only examining the values of $(\hat{g}_i, \hat{g}_{i+1})$ which are produced by the case of using u_k. If either $\hat{g}_{i+1}$ or $\hat{g}_i$ falls off the boundary of 0 or 255, we regard the block to have the possibility of falling-off, and abandon the block for embedding data.

In addition, the inverse calculation in Eq. (2) is designed in such a way that it satisfies the following property:

$$f((g_i,g_{i+1}),\mathrm{m}) = f(f((g_i,g_{i+1}),m'),m'')$$
$$\text{for } m = m' + m'' \qquad (3)$$

This equation means that the inverse calculation can proceed directly or progressively. This property is useful for judging the existence of embedded data in each block in the data recovering process.

An illustration of the data embedding process is shown in Fig.2. In the figure, the gray values of a sample two-pixel block are assumed to be (50,65). The difference value is 15, which is in the range of 8 through 23. The width of the range is $16 = 2^4$, which means that a difference value in the range can be used to embed four bits of secret data. Assume that the four leading bits of the secret data are 1010. The value of this bit stream is 10. It is added to the lower bound value 8 of the range to yield the new difference value 18. Finally, by Eq. (2) the values (48, 66) are computed for use as the gray values in the stego-image. Note that 66 - 48 = 18.

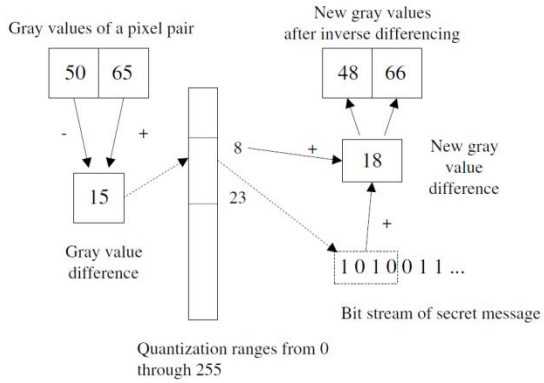


Fig 2: An illustration of the data embedding process

D. Process of recovering embedded data from stego images

To extract data, we do exactly the same procedures done in data embedding. The stego image is divided into

Bz x *Bz* blocks and the blocks are then rotated by random degrees based on the secret key key_1. The resulting image is rearranged as a row vector V^1. Finally, we get the embedding units by dividing V^1 into non overlapping blocks with two consecutive pixels.

he process of extracting the embedded message proceeds by using the seed of the pseudorandom scheme to produce the same traversing order for visiting the two-pixel blocks as in the embedding process. Each time we visit a two-pixel block in the stego-image, we apply the same falling-off-boundary checking as mentioned previously to the block to find out whether the block was used or not in the embedding process. Assume that the block in the stego-image has the gray values (g^*_i, g^*_{i+1}) and that the difference d of the two gray values is with index k. We apply the falling-off-boundary checking process to (g^*_i, g^*_{i+1}) by using $f((g^*_i, g^*_{i+1}), u_k\text{-}d^*)$.

We now want to prove that the resulting $(\hat{g}^*_i, \hat{g}^*_{i+1})$ computed from $f((g^*_i, g^*_{i+1}), u_k\text{-}d^*))$ are identical to the gray values $(\hat{g}_i, \hat{g}_{i+1})$ which were computed by $f((g_i, g_{i+1}), u_k - d)$ in the embedding process. The proof is as follows. First,

$$(\hat{g}_i, \hat{g}_{i+1}) = f((g_i, g_{i+1}), u_k\text{-}d))$$
$$= f((g_i, g_{i+1}), d^* - d + u_k - d^*) \qquad (4)$$

By Eq. (3), the above result can be transformed further to be

$$f(f((g_i, g_{i+1}), d^* - d + u_k - d^*) = f(f((g_i, g_{i+1}), d^* - d), u_k - d^*))$$
$$= f((g^*_i, g^*_{i+1}), u_k - d^*)) = (g^*_i, g^*_{i+1}) \quad (5)$$

This completes the proof.

The above property shows that the results of both of the falling-off-boundary checking processes, one in data embedding and the other in data recovery, are identical. This in turn implies that if either of the gray values of the computed values (g^*_i, g^*_{i+1}) falls off the boundaries of the range [0, 255], it means that the current block was not used for embedding data, or that the block was abandoned in the embedding process. On the contrary, if both of the values (g^*_i, g^*_{i+1}) do not falloff the range, it means that some data was embedded in the block. The value b, which was embedded in this two-pixel block, is then extracted out using the equation

$$b = \begin{cases} d^* - l_k & \text{for } d^* \geqslant 0; \\ -d^* - l_k & \text{for } d^* < 0. \end{cases} \quad (6)$$

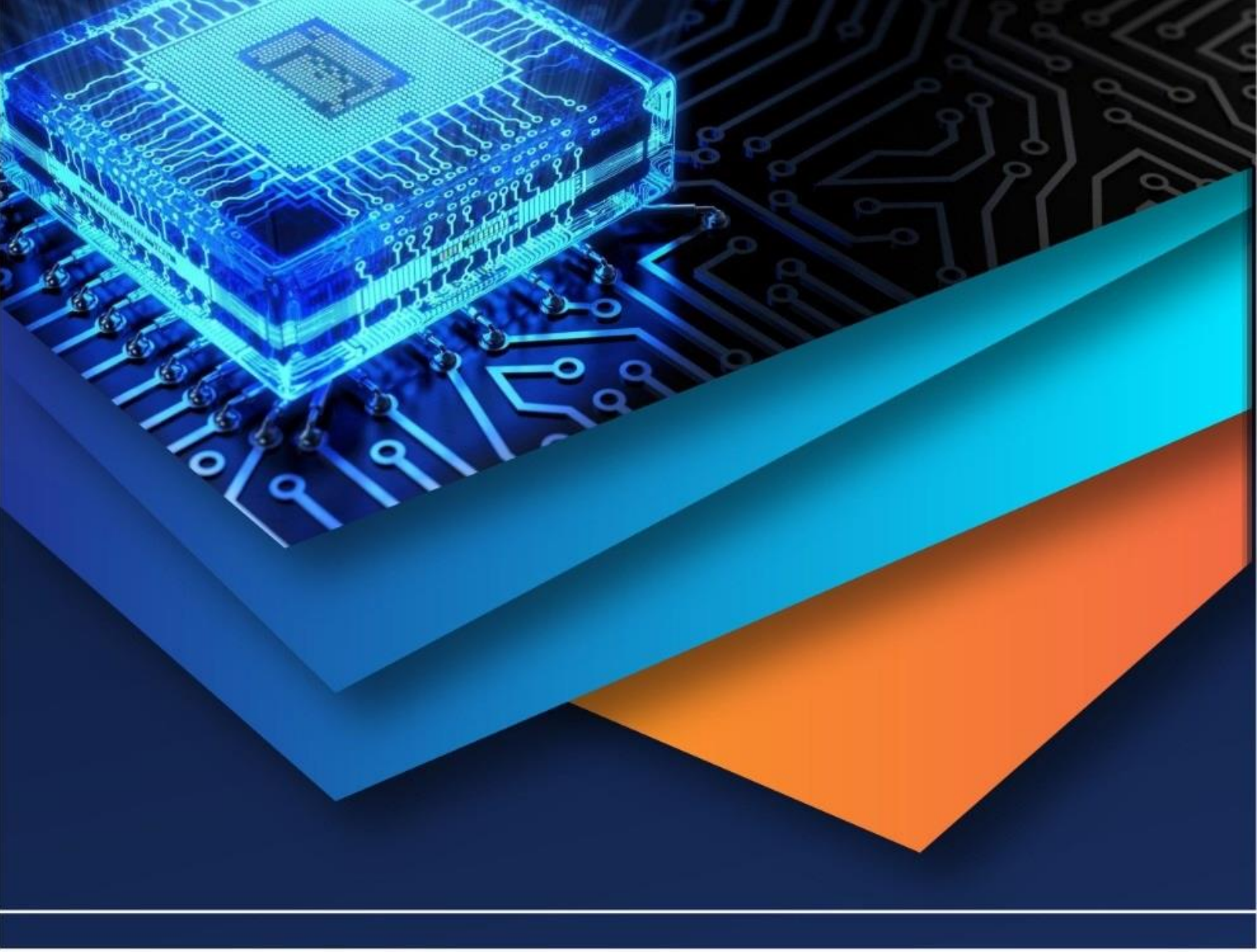
Note that in the recovery of the secret messagefrom the stego-image using the previously described extraction process, there is no need of referencing the cover image.

IV. CONCLUSION

A new and efficient steganographic method for embedding secret messages into images without producing noticeable changes in the stego image has been proposed. There is no need of referencing the original image when extracting the embedded data from a stego-image. The method utilizes the characteristic of the human visions sensitivity to gray value variations. Secret data are embedded into a cover image by replacing the difference values of the two-pixel blocks of the cover image with similar ones in which bits of embedded data are included. Also the addition of edge analysis before data embedding to extract the candidate region ensures better results. The method not only provides a better way for embedding large amounts of data into cover images with imperceptions, but also offers an easy way to accomplish secrecy.

REFERENCES

[1] T. Filler, J. Judas, and J. Fridrich, “Minimizing embedding impact in steganography using trellis-coded quantization,” in Proc. SPIE, Media Forensics and Security, 2010, vol. 7541, DOI: 10.1117/12.838002.
[2] S. Lyu and H. Farid, “Steganalysis using higher-order image statistics,”IEEE Trans. Inf. Forensics Security, vol. 1, no. 1, pp. 111–119, Mar.2006.
[3] E Lin and E Delp, “A Review of Data Hiding in Digital Images” CERIAS Tech Report 2001-139
[4] J. Mielikainen, “LSB matching revisited,” IEEE Signal Process. Lett., vol. 13, no. 5, pp. 285–287, May 2006.
[5] C. K. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” Pattern Recognition, vol. 37, no. 3, pp. 469–474, 2004.
[6] X. Zhang and S. Wang, “Efficient steganographic embedding by exploiting modification direction,” IEEE Commun. Letter., vol. 10, no. 11, pp. 781–783, Nov. 200
[7] R.M. Chao, H. C. Wu, C. C. Lee, and Y. P. Chu, “A novel image data hiding scheme with diamond encoding,” EURASIP J. Inf. Security, vol. 2009, 2009, DOI: 10.1155/2009/658047, Article ID 658047.
[8] J. Wang, Y. Sun, H. Xu, K. Chen, H. J. Kim, and S. H. Joo, “An improved section-wise exploiting modification direction method,” Signal Process., vol. 90, no. 11, pp. 2954–2964, 2010.
[9] M. Nosrati, R. Karimi, H. Nosrati, and A. Nosrati, “Embedding stego-text in cover images using linked list concepts and LSB technique”, Journal of American Science, Vol. 7, No. 6, 2011, pp. 97-100.
[10] Weiqi Luo, Fangjun Huang, “Edge Adaptive Image Steganography Based on LSB Matching Revisited” , IEEE transactions on infoemation forensics and security, Vol. 5, No. 2, June 2010
[11] W. Hong and T. S. Chen, “Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism,” J. Vis. Commun. Image Represent., vol. 22, no. 2, pp. 131–140, 2011.
[12] Wien Hong ,Tung-Shou Chen, “A Novel Data Embedding Method Using Adaptive Pixel Pair Matching”, IEEE transactions on information forensics and security, Vol. 7, No. 1, February 2012.
[13] K. Hempstalk, “Hiding behind corners: Using edges in images for better steganography,” in Proc. Computing Women’s Congress,Hamilton, New Zealand, 2006.
[14] Da-Chun Wu, Wen-Hsiang Tsai, “A steganographic method for images by pixel-value differencing”, in Pattern Recognition Letters 24 (2003) 1613–1626, Elsevier



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY