New Image Steganography Method Based on K-means Clustering

Ismail KICH
Research Team MSISI – LaRIT
Faculty of Sciences, IbnTofail University
Kenitra, Morocco
kich.ismail.25@gmail.com

El Bachir AMEUR
Research Team MSISI – LaRIT
Faculty of Sciences, IbnTofail University
Kenitra, Morocco
ameurelbachir@yahoo.fr

Abdelghani SOUHAR
Research Team MSISI – LaRIT
Faculty of Sciences, IbnTofail University
Kenitra, Morocco
houssouhar@gmail.com

ABSTRACT

In this paper, a new image steganography method based on k-means clustering for embedding secret messages into a gray image is proposed. The proposed methodology is a combination of two techniques, image clustering and Least Significant Bits replacement. In the embedding process, a cover image is segmented into clusters using the K-means clustering algorithm. Each cluster is partitioned into two regions, smooth and complex, only smooth regions may contain the secret data. In this manner, degradation of the Stego image quality is imperceptible to the human eye. For better protection of secret message, a pseudo-random key mechanism is coupled to the implementation of the LSB method. The secret data can be recovered directly from the image Stego without reference to cover-image. The experimental results show that the proposed method has a high image quality and a good embedding capacity.

KEYWORDS

Clustering, K-means, Image security, Steganography, Data Hiding.

1 INTRODUCTION

The explosion of communication networks, the arrival of the internet and the enthusiasm of the population for new technologies have created a large flow of information (image, text, sound, video ...) in the whole world, thus ensuring safety during data transfer is a primary task and must be applied and developed.

Steganography is the art of hiding and transmitting data through apparently innocuous carriers to conceal the existence of data [1]. The word is derived from the Greek words “stegos” meaning cover and “grafia” meaning writing defining it as “covered writing”. Unlike cryptography used to secure data by changing the way of arrangement, the steganography hides the message by enclosing it in media which can be distributed and used normally [2]. Many different file formats can be used to hide data, but digital images are the most popular because of their frequency on the Internet [3].

Image steganography exploits the limitations of Human Visual System. Human eye cannot detect the variation in luminance of color vectors at high frequency side of the visual spectrum. In the literature, many techniques about data hiding have been proposed. One of the most famous techniques is based on manipulating the Least-Significant-Bits (LSB) of host image pixels by replacing the LSB bits by the secret message bits [4], given the simplicity of its implementation and also the large capacities of data that can be inserted with it.

Various techniques have been implemented based on the spatial domain or the frequency domain approaches to perform image Steganography. Criteria such as imperceptibility, capacity, robustness against various attacks etc. are taken into account when developing such algorithms [5-8].

To increase the imperceptibility of the Stego image, Wu and Tsai in [9] propose a novel steganography method that uses the difference value between two neighboring pixels to determine how many secret bits should be embedded. Wu et al. Proposed to combine the PVD technique and the base decomposition scheme [10]. Other methods use the same principle combined with methods based on LSB algorithm to reduce Mean Squared Error (MSE) values and increase efficiency [11-13].

Interpolation is one of the techniques of image processing applied in reversible data hiding. To improve the embedding capacity and provide an imperceptible visual quality, a novel steganography method based on interpolation methods are presented in [14-17].

Clustering is a method of grouping data objects (pixels) into different groups, such that similar data objects belong to the same group and dissimilar data objects to different clusters. Clustering on image helps to get different sets of useful data [18-20].

BDCA'17, March 29-30, 2017, Tetouan, Morocco
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4852-2/17/03 $15.00
http://dx.doi.org/10.1145/3090354.3090432

In this paper, we propose a new image steganography method based on k-means clustering to provide a better imperceptibility for the Stego-image and better security for hidden data. The Cover-image is clustered into k cluster, the number of clusters k and k-means algorithm initialization centroids are defined from its histogram. In this way, each cluster will contain pixels that have similar values. The resulting clusters are ranked in descending order per their cardinal. Changing pixel values by applying the LSB method may cause a cluster change to which these pixels belong. To remedy this problem each cluster is then partitioned into two regions, smooth and complex. Smooth regions are those that will incorporate the secret message while complex regions do not receive any secret information. The data hiding process using a random key to increase security is made in smooth regions of each cluster, in descending order per the secret message size. Thus, we get a Stego-image that will be sent to the other user.

This paper is organized as the following: Section II introduces the k-means clustering algorithm. The description of simple LSB method is presented in section III, and the proposed image steganography method is described in section IV. In section V a case study is presented. In section VI we present the experimentation of our approach before ending with a conclusion and future works.

2 K-MEANS ALGORITHM

K-means is one of the most popular methods used to apply the clustering process. It classifies a given set of data into k number of disjoint clusters. The K-means algorithm consists of two separate phases. In the first phase, it calculates the k centroid and in the second phase, it takes each point to the cluster, which has nearest centroid from the respective data point. There are different methods to define the distance of the nearest centroid and one of the most used methods is Euclidean distance. Once the grouping is done it recalculates the new centroid of each cluster and based on that centroid, a new Euclidean distance is calculated between each center and each data point and assigns the points in the cluster, which have minimum Euclidean distance. Each cluster in the partition is defined by its member objects and by its centroid. The centroid for each cluster is the point to which the sum of distances from all the objects in that cluster is minimized.

Let us consider an image with resolution of $L{\times}C$ pixels to be clustered into k clusters. Let $p_i, i \in \{0, \ldots, L{\times}C\}$ be an input pixel to be clustered and G_k is the k-th centroid of k-th cluster C_k. The algorithm for K-means clustering is as follows: (see Fig. 1).

1. Initialize number of cluster k and centroids.
2. For each pixel of an image p_i, calculate the Euclidean distance d_k between the pixel and centroid G_k of k-th cluster C_k
$$d_k = d(p_i, G_k) = \|p_i - G_k\|$$
3. Assign all the pixels to the nearest centroid based on distance d_k using:
$$\underset{k}{\arg}\ \min \mathrm{d}(p_i,\ G_k)$$
4. After all pixels, have been assigned, recalculate new position of the centroids using:
$$G_k = \frac{1}{N_k} \sum_{i \in C_k} p_i$$
Where
$$N_k = card(C_k)$$
5. Repeat the process until it satisfies the tolerance or error value.
6. Reshape the cluster pixels into image.

Although k-means has the great advantage of being easy to implement, it has some drawbacks. The quality of the final clustering results depends on the arbitrary selection of initial centroid. So, if the initial centroid is randomly chosen, it will get different result for different initial centers. So, the initial center will be carefully chosen so that we get our desire segmentation. And, computational complexity is another term which we need to consider while designing the k-means clustering. It relies on the number of data elements, number of clusters and number of iterations.

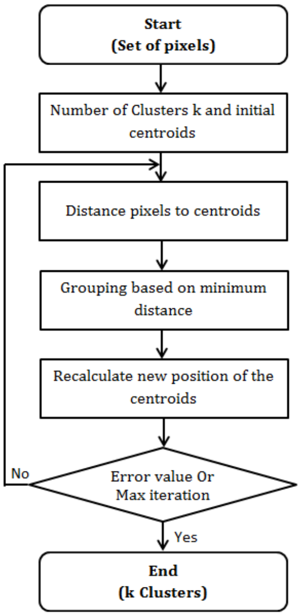
Start
(Set of pixels)
Number of Clusters k and initial centroids
Distance pixels to centroids
Grouping based on minimum distance
Recalculate new position of the centroids
No
Error value Or Max iteration
Yes
End
(k Clusters)

Figure 1: K-means Flowchart

3 SIMPLE LSB METHOD

In this section, we describe the general operations applied by simple LSB method for hiding data.

Let I be the original 8-bit gray scale cover-image of $L{\times}C$ pixels represented as:

$$I = \{x_{ij} \mid 0 \le i \le L, 0 \le j \le C,\ x_{ij} \in \{0, \ldots, 255\}\}$$

M is the n-bit-Stream of secret message represented as:

$$\mathrm{M} = \{m_i \mid 0 \le i \le n, \quad m_i \in \{0,1\}\}$$

Suppose that we use the N-rightmost LSBs of the cover-image I to embedding the n-bit-Stream secret message M. Firstly, the secret message M is rearranged to form a conceptually N-bit set of values M' represented as:

$$M' = \{m_i^{'} \mid 0 \le i \le n', \; m_i^{'} \in \{0,1,\ldots,2^N - 1\}\}.$$

$$\text{Where} \quad n' < L \times C$$

The mapping between the n-bit-Stream secret message $M = \{m_i\}$ and the embedded message $M' = \{m'_i\}$ can be defined as follows:

$$m_i' = \sum_{j=0}^{N-1} m_{i\times N+j} \times 2^{N-1-j}$$

Secondly, a subset of n' pixels $\{p_1, p_2, \ldots\ldots, p_{n'}\}$ is chosen from the cover-image I in a predefined sequence. The embedding process is completed by replacing the N LSBs of p_i by m_i'. Mathematically, the pixel value p_i of the chosen pixel for storing the N-bit message m_i' is modified to form the stego-image p_i' as follows:

$$p_i' = p_i - p_i \bmod 2^N + m_i'$$

In the extraction process, the embedded data can be easily extracted from the image-Stego S without referring to the original image I. Using the same sequence as in the embedding process, the set of pixels $\{p'_1, p'_2, \ldots\ldots, p'_{n'}\}$ storing the secret message bits are selected from the Stego-image. The N LSBs of the selected pixels are extracted and lined up to reconstruct the secret message bits. Mathematically, the embedded message bits m_i' can be recovered by $m_i' = p'_i \bmod 2^N$.

4 PROPOSED SYSTEM

In this section, new image steganography method based on k-means clustering is proposed.

The host image is segmented into k clusters, and the resulting clusters are then ranked from the biggest to the smallest per the pixels they contain. To increase the time calculated for the convergence of the k-means algorithm, the choice of the number of clusters k and initialization clustering centroids are defined from the histogram of the host image.

In descending order of their cardinal, the insertion is done in each cluster in per secret data size, the selected cluster is partitioned into two regions, smooth and complex. The smooth region is one that will contain the secret data while the complex region will not include any data; in general, the number of pixels of the complex region is minimal comparing to the first. The aim of partitioning the chosen cluster is to exclude pixels that can change their clusters and provide more distortion of quality of Stego-image during the embedding process. That is to say, the smooth region is the internal part of the cluster, it consists of the pixels that do not change their cluster and do not affect the image quality if they have changed their values during use of the LSB replacement method. While the complex region represents the contour of the cluster composed of pixels, which may change their cluster and affect remarkably the imperceptibility of the image in case of change of values. The user per the message size it wants to hide sets the number of bit N of each pixel used to embed the secret data.

The partitioning of a cluster into smooth and complex regions is based on the following procedure:

Let C_k, the k clusters of host image resulting by using K-means algorithm. To partitioning each cluster C_k into two regions smooth and complex, we develop the following function:

$$h_k(p_i) = \begin{cases} 1, & \underset{k,T}{\arg}\, \min \mathrm{d}(p_i', G_k) = C_k \\ 0, & elsewhere \end{cases}$$

Where:

p_i: The value of a pixel in cluster C_i;

$p_i' = p_i - p_i \bmod 2^N \pm T$;

T: Cluster partitioning threshold linked to the number N of bits used to embed the secret data in each pixel $T = 2^N - 1$;

k : The number of clusters;

G_k : The centroid of the cluster C_k;

$d(p_i', G_k)$: Euclidean distance between p'_i and centroid of cluster C_k;

$\underset{k,T}{\arg}\, \min \mathrm{d}(p_i', G_k)$: return the cluster belongs p_i'.

Note that we must calculate h_k for the two cases of p_i', by adding T in first case, and subtracting it in the second.

With this function, the pixels valued 1 in C_i represent the smooth pixel, and valued 0, the complex one.

To further increase the security level, a random-key is used as seed for the Pseudo-Random Number Generator (KEY), the data are embedded in the smooth region of the first cluster, and then we move to the second cluster when space in the current cluster is not enough by order according to the ranked clusters.

Finally, the Stego image is subjected to another processing to integrate the final centroids values resulting from the algorithm of k-means and the key (KEY) before sending it to the recipient. This two information will be used in the extraction phase. The steps in the proposed system are explained in Fig. 2.

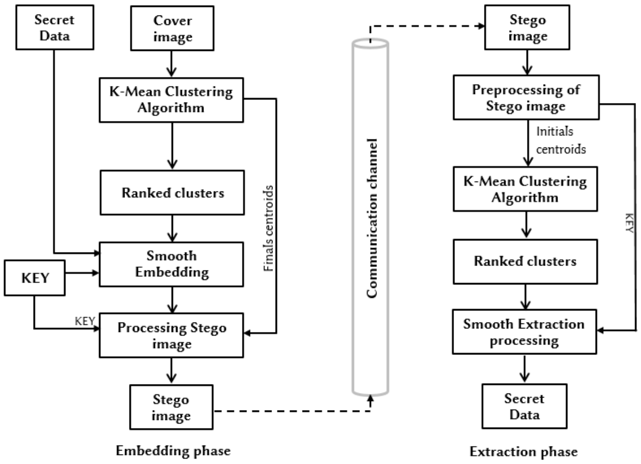

Figure 2: Embedding and Extraction Algorithm

The embedding phase and the extraction phase will follow the same steps except that the inputs and outputs of the phases are different. The first phase has as input the cover image, the secret data and the key (KEY). While the second phase uses the

Stego image preprocessed by the recipient to deduce the K-means algorithm initialization centroids and the key (KEY) before starting the process of extracting the secret data.

5 CASE STUDY

For the case study, the most popular image Lena is taken and proposed system is explained using this image. The steps implemented in this approach are explained and shown below.

5.1 Embedding Algorithm

Step 1: The cover image and corresponding histogram are as shown in the Fig. 3:

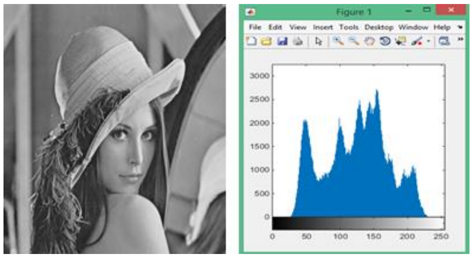

Figure 3: Input image and corresponding Histogram.

The values of the initial centroid used to launch the k-means algorithm are computed automatically after choosing the number k of the cluster from the histogram of the cover image. k=4 and G_1=50, G_2=100, G_3=150, G_4=200.

Step 2: The cover image is segmented using K-Means Clustering, where k = 4 and initialization centroid are cited above. The result is figured in Fig. 4:

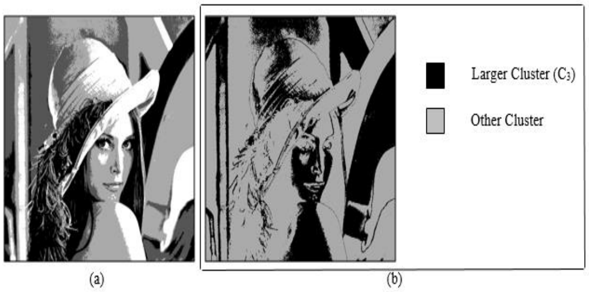

Figure 4: All clusters (a), Larger Cluster C_3 (b).

Step 3: After clustering, the resulted clusters are ranked in descending order based on their cardinal. The clusters are segmented into too regions, smooth and complex. The smooth region is chosen to embed the secret data.

The pixels of the smooth region keep their affiliations in the same cluster after they are modified, this modification also does not significantly distort the image quality. The goal is to have the same clustering of the cover image during the embedding and extraction of secret data, and keep a good Stego image imperceptibility. To increase the capacity of the proposed method we use different values of the last significant bits N for embedding secret data.

The Fig. 5 shows the first cluster colored black selected to insert the data (c), the smooth and complex regions for different values of $T = 2^N - 1$ in the same cluster, (d), (e), (f).

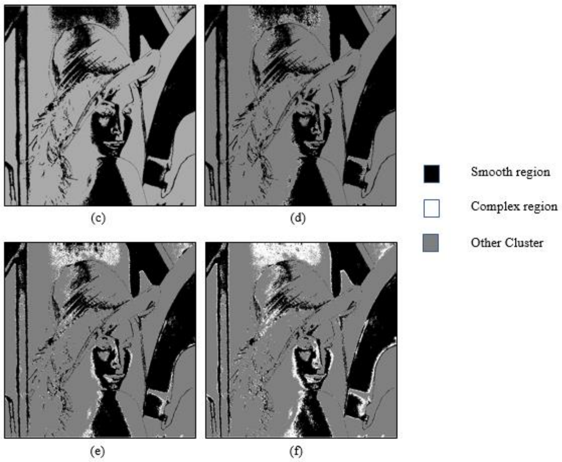

Figure 5: Larger Cluster C3(c), Smooth and Complex region C3 for T=1 (d), T=3 (e), T=7(f).

Step 4: Data insertion, the message is embedded using the LSB technique strengthened by a key (KEY) in the smooth regions of clusters, and as result of this step, we obtain the Stego-image that will be communicated to other users.

5.2 Extraction Algorithm

The approach taken during the extraction phase of secret data remains the same as the integration phase, except that the extraction is done directly from the Stego image without resort to Cover-image. The Stego image must be pretreated in order to obtain initialization centroids of K-mean algorithm and the key (KEY) as the input of LSB algorithm to retrieve the secret data.

6 EXPERIMENT RESULTS

In this section, we present and discuss the experimental results of the proposed method. Standard test images, of size 512×512 obtained from USC image database [21] were employed as the cover images, as shown in Fig. 6. These color images were transformed into 8-bit gray scale images if they were originally in color format.

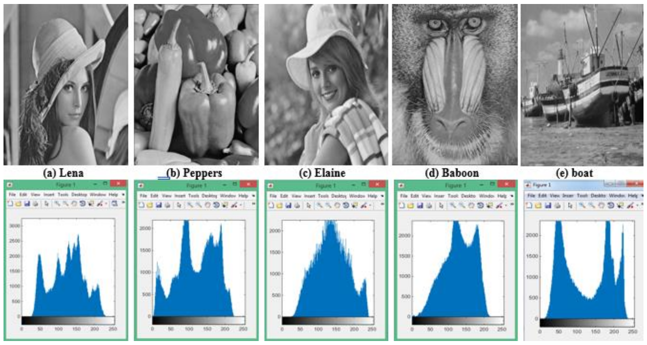

Figure 6: Test images and corresponding Histograms.

To appraise the quality of Stego image, Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) are estimated. Formulas below show the procedure for calculating MSE and PSNR for an $L \times C$ grayscale image respectively.

$$MSE = \frac{1}{L * C} \sum_{i=1}^{L} \sum_{j=1}^{C} [p_{i,j} - p'_{i,j}]^2$$

$$PSNR = 10 * log_{10} \left(\frac{255^2}{MSE} \right)$$

Where $p_{i,j}$ represent the pixels of Cover image and $p'_{i,j}$ the Stego image.

For examining our proposed method, we use different size of random messages as secret data to be embedded in cover image and extracted from Stego image. Table 1 show the PSNR of Stego image for different size of secret data, using different threshold values $T = 2^N - 1$ where N is the number of LSB used to hide information in each pixel. We note also that the hidden and extracted message are equal in the various tests performed.

Table 1: Variation of PSNR for different images and sizes of secret data using different value of T

Cover image	K	T	500 byte	1000 byte	2500 byte
				PSNR	
Lena	4	1	69,2907	66,2826	62,3175
		3	65,5429	62,6412	58,5773
		7	61.0251	58.0762	54.1865
Peppers	3	1	69,3168	66,3054	62,3002
		3	65.2371	62.4822	58.5052
		7	61.2795	57.9368	54.2794
Elaine	2	1	69,2691	66,2192	62,2850
		3	65.3728	62.3179	58.4698
		7	61.0007	58.1718	53.9810
Baboon	2	1	69,1965	66,2934	62,3179
		3	65.5658	62.2618	58.5548
		7	61.0734	57.8333	54.1011
Boat	3	1	69,2327	66,1947	62,2803
		3	65.1645	62.2047	58.3445
		7	61.1042	58.0387	54.1511

Fig. 7 shows PSNR's evolution per the secret message size that we can hide inside the Lena test image for different values of T.

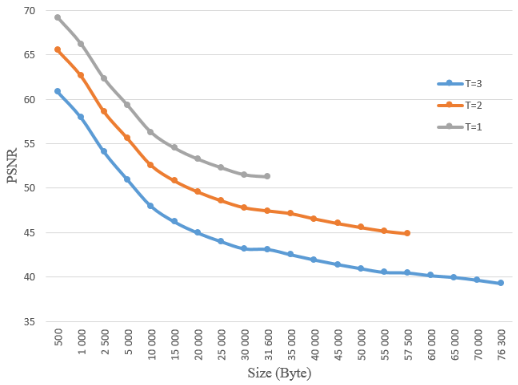

Figure 7: PSNR's evolution with different values of T.

The results show that by increasing the threshold T, large amounts of data can be embedded into the cover image, thereby ensuring a very high level of secrete of transmitted data and also a good quality of Stego image.

Finally, we compare our method to those based on interpolation, specifically the method of Jung and Yoo [17] and that of Hu and Li [15]. The Table 2 shows the result of the proposed method compared to semi-reversible data hiding method of Jung et Yoo for 2-LSB and 3-LSB substitution. Note that the capacity represents number of maximal bits used for embedding secret data. In our proposed method based on k-means clustering algorithm, the secret data is embedded in smooth region of each cluster. The capacity is given by the following equation:

$$Capacity = N \times \sum_{j=1}^{k} Card(SmoothRegion(j))$$

Where SmoothRegion(j) is smooth region of the j^th cluster.

Table. 2: A comparison of image quality and capacity with Jung and Yoo's method for N=2 and N=3.

Cover image	N	Jung and Yoo's method [17]		Proposed method	
		Capacity (bit)	PSNR	Capacity (bit)	PSNR
Lena	2	393216	43.95	460898	44.88
	3	589824	37.56	610683	39.28

Baboon	2	393216	43.94	488140	44.62
	3	589824	37.54	684861	38.77
Peppers	2	393216	43.93	506236	44.47
	3	589824	37.50	702492	38.68
Boat	2	393216	43.93	478560	44.71
	3	589824	37.48	595614	39.36

The following table shows the results of comparison with the interpolation method based on Maximizing the difference values between Neighbouring Pixels (IMNP) of Hu and Li [15].

Table. 3: A comparison of image quality and capacity with IMNP method.

Cover image	IMNP method [15]		Proposed method	
	Capacity(bpp)	PSNR	Capacity(bpp)	PSNR
Lena	1.6945	34.69	1.7581	44.88
Baboon	1.8483	30.14	1.8621	44.62
Boat	2.0437	32.98	2.2720	39.36

The PSNR values shown in the results of the comparisons presented in tables above show that our method by contribution to the IMNP method, Yang and Yoo's method can improve the quality of Stego image while ensuring a better security of transferred data. Note that we used N = 2 to incorporate secret data into the Lena and Baboon images and N = 3 for Boat.

7 CONCLUSION

To increase the security level of data to be transmitted, using steganography, we propose a new steganography method through gray images based on K-means Clustering algorithm and classical Least Significant Bits (LSB) technique. We apply the k-means algorithm to divide the image into several groups. Our method ranks the clusters in descending order of their cardinal, and uses smooth pixels of each cluster for the embedding secret data. According to the size of the secret data, the user can select the threshold T that allows using one bit or more of each pixel of the cover image to hide the secret data. The experimental results compared to the IMNP [15] method and that of Jung and Yoo [17]show a high image quality and a good integration capacity. In future work, this method can be extended to other clustering method.

REFERENCES

[1] N. Provos, P. Honeyman, Hide, and seek, 2007. an introduction to steganography. *IEEE Security and Privacy Magazine* 1 (2003), 32–44.

[2] A. Cheddad. J. Condell, K. Curran, P.M. Kevitt, 2010. Digital image steganography: survey and analysis of current methods. *Signal Processing* 90 (3) (2010) 727–752.

[3] J. Fridrich. M. Goljan, and R. Du, P.M. Kevitt, 2001. Reliable detection of LSB steganography in color and grayscale images. *In Proceedings of the workshop on multimedia and security: new challenges. ACM*, 2001, 27–30.

[4] D.W. Bender. N.M. Gruhl, and A. Lu, 1996. Techniques for data hiding, *IBM Syst. J.* 35 (1996), 313-316.

[5] Y. Taouil. E.B. Ameur, and M.T. Belghiti, 2016. New image steganography method based on Haar Discrete Wavelet Transform. *Advances in intelligent systems and computing*, Vol. 520 (2016), 287-297.

[6] C.K. Chan and L.M. Chen, 2004. Hiding data in images by simple LSB substitution. *Pattern Recognition*, Vol. 37, No. 3 (2004), 496-474.

[7] R.Z Wang, C.F. Lin, and J.C. Lin, 2001. Image hiding by optimal LSB substitution and genetic algorithm. *Pattern Recognition* 34, (2001), 671–683.

[8] I.C. Lin, Y.F. Lin, and C.M. Wang, 2009. Hiding data in spatial domain images with distortion tolerance, *Comput. Stand. Inter.* 31 (2) (2009), 458–464.

[9] D.C. Wu, and W.H. Tsai, 2009. A steganographic method for images by pixel-value differencing, *Pattern Recognit. Lett.* 24 (9–10) (2003), 1613–1626.

[10] N.I Wu, KC. Wu, and C.M. Wang, 2012. Exploring pixel-value differencing and base decomposition for low distortion data embedding, *Appl Soft Comput* 12 (2012), 942–960.

[11] C.M. Wang, N.I. Wu, M.S. Hwang, and C.S. Tsai, 2008. A high quality steganographic method with pixel-value differencing and modulus function, *Journal of Systems and Software* 81 (2008), 150–158.

[12] X. Liao, Q. Wena, and J. Zhang, 2011. A steganographic method for digital images with four-pixel differencing and modified LSB substitution, *J. Vis. Commun. Image R.* 22 (2011), 1–8.

[13] Y.R. Park, H.H. Kang, , S.U. Shin, and K.R. Kwon, 2005. A Steganographic Scheme in Digital Images Using Information of Neighboring Pixels, *Springer-Verlag.* Vol. 3612, Berlin, Germany (2005), 962–967.

[14] W. Hong and T.S. Chen, 2011. Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism. *J. Vis. Commun. Image R* 22 (2011), 131–140.

[15] Jie Hu, and Tianrui. Li, 2015. Reversible steganography using extended image interpolation technique, *Computers and Electrical Engineering*, Vol. 46 Issue C (2015), 447–455.

[16] A. Benhfid, E.B. Ameur, and Y. Taouil, 2016. High capacity data hiding methods based on spline interpolation, In *proceeding ICMCS'16, Marrakech Morocco* (2016).

[17] Ki-Hyun Jung and Kee-Young Yoo, 2014. Steganographic method based on interpolation and LSB substitution of digital images, *Journal Multimedia Tools and Applications*, Vol. 74 Issue 6 (2014), 2143–2155.

[18] N. Dhanachandra, K. Manglem, and Y.J. Chanu, 2015. Image Segmentation using K-means Clustering Algorithm and Subtractive Clustering Algorithm, *Procedia Computer Science* 54 (2015), 764–771.

[19] C. Singh, and G. Deep, 2013. Cluster Based Image Steganography UsingPattern Matching, *International Journal of Emerging Trends & Technology in Computer Science*, Vol. 2, Issue 4 (2013).

[20] Nutan C. Malekar, and R.R. Sedamkar, 2014. Novel K-Means Clustering Approach for CompressingHyperspectral Image, *International Journal of Advancements in Research & Technology*, Vol. 3, Issue 1 (2014).

[21] Image database. Available from: http://sipi.usc.edu/database.