Cover Optimization for Image in Image Steganography

Nidhal K. El Abbadi
University of Kufa
Najaf, Iraq

Abstract

This paper develops techniques for discriminating between images which used as steganography cover. Algorithm is based on the hypothesis that a particular message embedding scheme leaves statistical evidence or structure that can be exploited for detection with the aid of proper selection of image features analysis. We pointed out the features of image that should be taken more seriously into account in the design of more successful steganography, weight for each of these features determined by using Analytic Hierarchy Process (**AHP**) which helps to maximize some of the features and gives weight according to the relation between these features. The proposed algorithm tested by using LSB image steganography, stego-image compared with the origin one which gives the promised results.

***Keywords**: steganography, features, AHP, information hiding, image.*

1. Introduction

Steganography is the art and science of hiding information by embedding data into media. Steganography (literally meaning covered writing) have been used since ancient time.

Electronic steganography techniques use digital ways of hiding and detecting processes. Normally the detection process is working inversely of the hiding process. Steganography is different from cryptography and watermarking although they all have overlapping usages in the information hiding processes. Steganography security hides the knowledge that there is information in the medium cover, where cryptography revels this knowledge but encodes the data as cipher-text and disputes decoding it without permission; i.e., cryptography concentrate the challenge on the decoding process while steganography adds the search of detecting if there is hidden information or not. Watermarking is different from steganography in its main goal. Watermarking aim is to protect the cover medium from any modification with no real emphasis on secrecy. It can be observed as steganography that is concentrating on high robustness and very low or almost no security [6].

Steganography techniques use different carriers (cover medium in digital format) to hide the data, these carriers may be network packets, hard drive, amateur radio waves, or generally any computer file types such as text, image, audio and video. Restrictions and regulations are thought of in using steganography due to the threat from law and rights enforcing agencies and the need of organizations aiming to secure their information. Many easy to use steganography tools are available to hide secret messages on one side of communication and detect hidden info. on the other side. Steganography uses cover to embedded secret data, this cover chooses randomly and for the same secret data every one can choose different cover without a prior knowledge which one is better, because there are no rules or measurements use for choosing suitable cover.

In this work, we propose many features that can be used to choose the best cover among many suggested covers for embedded secret data (image in image steganography). It also used the Analytical Hierarchy Process (AHP) to determine the weight for each feature. Unfortunately there are no studies about this problem. As best of my knowledge there are only two studies related to choosing cover, the first one presented by Mehdi [6] which studied the cover selection problem through three scenarios in which the secret data either no knowledge, partial knowledge, or full knowledge of the steganalysis technique. Hedieh [4], also presented a technique to compute steganography capacity as a property for image cover selection. This technique used different steganlyzer units, which help to determine the maximum size of embedded that can embedded in cover.

2. Methodology

The aim of this algorithm is to find the best cover for an embedded secret data, it focus on image in image steganography, for that many images features chooses to be scale to select best cover among many suggested covers, the weight for each feature can achieve by using (AHP method). These features will be modified in a way suitable with the aim of this paper. The features suggested to use are:

{Note: subscript (c): mean cover image, (e) mean embedded image, and (Pg) mean probability for color (g) in image = N (g)/M where:
N (g): number of pixels with color g, M: total number of pixels in image }

1. Entropy:

The entropy is a measure of image information content, which tells us how many bits we need to code the image data, and is given by [2].

$$Entropy = -\sum_{g=0}^{L-1} P_{(g)} \log_2 [P_{(g)}] \quad \ldots\ldots \quad (1)$$

Where L: Number of color in image
As the pixel values in the image are distributed among more color level, the entropy increases.

$$0 \leq \text{entropy} \leq \log_2 (L)$$

Coding redundancy occurs when the data used to represent the image are not utilized in an optimal manner. For cover and embedded entropy it is better that

$$Ent_c \geq Ent_e$$

The number of colors (NC) used in cover should be more than number of colors in embedded.
Number of colors in image is

$$NC = 2^{entropy}$$

Max colors different in an image (256 colors) are
$NC_c - NC_e$ equal to 256-1=255
Then the percent of difference in the number of colors (DNC) is

$$ENT = ((NC_c - NC_e)/255)*100 \quad \ldots\ldots\ldots\ldots \quad (2)$$

Note if $NC_e > NC_c$ then DNC will be negative and subtracted from final result.

2. Capacity:

This term refers to the amount of data that can be hidden in the medium. It is defined as "*the maximum message size that can be embedded subject to certain constraints*"[7].
There are restrictions of data rate that can be embedded in a certain image. The worst case of embedded data is 1 bit in each byte (8 bits) as in LSB which represents (12.5%) of cover size as a maximum.
If the size of data embedded in the cover increased to more than the capacity of cover, then its transparency will be affected; i.e. with very high capacity, the steganography is not strong to keep transparent from eavesdroppers.

To check the capacity you should follow the following steps:

(a) $(size_e/size_c) \leq 0.125$
b) if the result in step (a) is false then we calculate the percent of capacity compatibility (CC) between cover and embedded is

$$CC = 100 - ((Size_e/size_c)/0.125)*100 \quad \ldots\ldots \quad (3)$$

3) Mean:

The mean is the average value which tells us something about the general brightness of the image.
A bright image will have a high mean (more than 127) and dark image will have low mean.

$$Mean = \sum_r \sum_c I(r, c)/m$$

The max difference in mean is 255.
% of mean similarity (MS):

$$MS = 100 - ((abs(g_c' - g_e')/255)*100 \ldots\ldots (4)$$

Where: g': color value mean

4) Variance:

Which tells us something about the contrast, it describes the spread in the data, so a high contrast image will have a high variance, and a low contrast image will have a low variance [17].

$$V_{(g)} = \sqrt{\sum_{g=0}^{L-1} (g - g')^2 \, p(g)} \quad \ldots\ldots\ldots\ldots\ldots \quad (5)$$

Max variance is when there are just two colors one equal to zero and other equal 255, then the mean is equal to (127.5) and the max variance is (127.5).
It is recommended that V_e approach to zero.
Variance similarity (Vs) is calculated as a percent
%VS =

$$((V_c - V_e)/127.5)*100 \quad \ldots\ldots\ldots\ldots\ldots (6)$$

5) Histogram:

Histogram analysis may be required before embedding to prevent the histogram attack [8].
Histogram matching between cover and embedded is done by comparing each color in cover histogram with the corresponding color in embedded histogram, if the number of pixels at that color is more than number of pixels in embedded for the same color then counter increases with one.
% color matching (CM):

$$CM = (counter/256)*100 \ldots\ldots\ldots\ldots\ldots\ldots (7)$$

6) Energy:

The energy measurement tells us something about how the colors distributed [17].

$$Energy = \sum_{g=0}^{L-1} (P_{(g)})^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

The energy measurement has a maximum value of (1) for an image with one color.

The larger this value is the easier to compress the image data. Energy indicates the region of image with identical color value, increasing energy mean increasing the size of this region, and the capability of compression will be increased.

The best distribution is when all colors (g) have the same frequency. (x: number of pixels have the same color g)

$$Energy = \sum_{g=0}^{L-1} x^2 / (size_c)^2$$

$$= \sum_{g=0}^{L-1} x^2 / (x*256)^2$$

$$= 256 * x^2 / (x^2 *(256)^2) = 1 / 256$$

Well, this value of energy (1/ 256) represent (100%) of distribution. Then when the energy value increases, the energy percent will decrease (inverse relation)

% distribution (DS)

$$= 1 / (energy_c *256))*100 \ldots\ldots\ldots\ldots\ldots (9)$$

7) Robustness

Robustness (R) can only be achieved by redundant information encoding which will degrade the cover heavily and possibly alter probability distribution P_s.

An embedding algorithm will be consider a robust if the embedded message can be extracted after an image has been manipulated without being destroyed. The more randomness that exists in an image the more evenly the color levels distributed and the more bits per pixels are required to represent the data. This also correlates to information more randomness implies each individual value is less likely which means more information is contained in each pixel value so we need more bits to code each pixel value and more robustness. Best robustness is when

$(P = x/ size_c)$

$X = size_c/256$

$P = (size_c /256)/size_c=1/256$

$$Entropy = - \sum_{g=0}^{255} P_c (g) \log_2 P_c (g)$$

$$= - \sum_{g=0}^{255} (1/256) \log_2 (1/256)$$

$$= \log_2 (1/256)$$

$\%R= - (entropy_c / (\log_2 (1/256))*100$ This can be simplified as

$$\% R = (entropy_c/8)*100 \ldots\ldots\ldots\ldots\ldots (10)$$

8) Expected Secrecy

Secrecy is one of the most important criteria. The secrecy is the ability to hide information in cover image, and is determined as a magnitude (ε) by comparing the cover image and stego- image according to relative entropy [10].

D (Pc||Ps) =

$$\sum_{g=0}^{L-1} P_c (g) \log_2 (P_c (g) / P_s (g)) \ldots\ldots\ldots (11)$$

The relative entropy between two distributions is always non-negative, and is zero if and only if the distributions are equal. We modify this equation to get a new relation that can determine the expected secrecy (the worst secrecy) without needing the existence of stego or hiding algorithm.

If we use LSB then the number of bytes (NB) that should be modified in covering it equals the number of embedded bits. Then

$NB = size_e \times 8$

The number of bytes from each color in cover should be changed depending on probability for each color.

$Prop (g) = freq_{(g)} / size_c$

where: freq= means number of color (g)

The number of bytes change for each color will be:

$NB_{(g)} = 8 \times size \times (ferq_{(g)}/size_c)$

That means each color (g) in cover will reduce with quantity of NB (g) and will increase with quantity of $NB_{(g-1)}$

Then the number of bytes of color (g) in stego will be

a) When(g) odd

SNB (g) =

$$freq_{(g)} – NB_{(g)} + NB_{(g-1)} \ldots\ldots\ldots\ldots\ldots (12)$$

b) when (g) even

SNB (g) =

$$freq_{(g)} – NB_{(g)} + NB_{(g+1)} \ldots\ldots\ldots\ldots (13)$$

Then according to first equation

Estimated Secrecy =

$$\sum_{g=0}^{255} P_c (g) \log_2 (P_c (g) / P_s (g))$$

$$= \sum_{g=0}^{255} (freq_c (g) / size_c) \log_2 ((freq_c(g) / size_c) / (freq_s(g) / size_s))$$

If we know that $Size_c = Size_s$

Estimated secrecy (ES) =

$$(1/ size_c)\sum_{g=0}^{255} freq_c(g) \log_2 (freq_c (g) / SNB(g))$$

Percent will determined according to

$\varepsilon = 2^{-secrecy}$

$$\%es = \varepsilon *100 \ldots\ldots\ldots (14)$$

3. Analytic Hierarchy Process (AHP)

The Analytic Hierarchy Process (AHP) is a mathematical technique for multi-criteria decision making [11]. It enables people to make decisions involving many kinds of concerns including planning, setting priorities, selecting the best among a number of alternatives, and allocating resources. AHP uses for relative criticality weighting of indicators, and relative criticality weighting of evaluators.

The Analytic Hierarchy Process (AHP) is a structured technique for dealing with complex decisions. Rather than prescribing a "correct" decision, the AHP helps the decision makers find the one that best suits their needs and their understanding of the problem.

Based on mathematics and psychology, it was developed by Thomas L. Saaty in the 1970s and has been extensively studied and refined since then. The AHP provides a comprehensive and rational framework for structuring a decision problem, for representing and quantifying its elements, for relating those elements to overall goals, and for evaluating alternative solutions. It is used around the world in a wide variety of decision situations, in fields such as government, business, industry, healthcare, and education.

Several firms supply computer software to assist in using the process.

Users of the AHP first decompose their decision problem into a hierarchy of more easily comprehended sub-problems, each of which can be analyzed independently. The elements of the hierarchy can relate to any aspect of the decision problem tangible or intangible, carefully measured or roughly estimated, well or poorly understood anything at all that applies to the decision at hand.

Once the hierarchy is built, the decision makers systematically evaluate its various elements by comparing them to one another two at a time. In making the comparisons, the decision makers can use concrete data about the elements, or they can use their judgments about the elements' relative meaning and importance. It is the essence of the AHP that human judgments, and not just the underlying information, can be used in performing the evaluations [12].

The AHP converts these evaluations to numerical values that can be processed and compared over the entire range of the problem. A numerical weight or priority is derived for each element of the hierarchy, allowing diverse and often incommensurable elements to be compared to one another in a rational and consistent way. This capability distinguishes the AHP from other decision making techniques.

In the final step of the process, numerical priorities are calculated for each of the decision alternatives. These numbers represent the alternatives' relative ability to achieve the decision goal, so they allow a straightforward consideration of the various courses of action.

As can be seen in the material that follows, using the AHP involves the mathematical synthesis of numerous judgments about the decision problem at hand. It is not uncommon for these judgments to number in the dozens or even the hundreds. While the math can be done by hand or with a calculator, it is far more common to use one of several computerized methods for entering and synthesizing the judgments. The simplest of these involve standard spreadsheet software, while the most complex use custom software, often augmented by special devices for acquiring the judgments of decision makers gathered in a meeting room.

The procedure for using the AHP can be summarized as:

1. Model the problem as a hierarchy containing the decision goal, the alternatives for reaching it, and the criteria for evaluating the alternatives.
2. Establish priorities among the elements of the hierarchy by making a series of judgments based on pair-wise comparisons of the elements
3. Synthesize these judgments to yield a set of overall priorities for the hierarchy.
4. Check the consistency of the judgments.
5. Come to a final decision based on the results of this process.
6.

We conduct AHP in three steps:

1. Perform pair-wise comparisons
2. Assess consistency of pair-wise judgments
3. Compute the relative weights
4.

- **Pair Wise Comparisons**

AHP enables a person to make pair wise comparisons of importance between decision elements (e.g., *child indicators* influencing a parent indicator, *evaluators* evaluating a leaf indicator) with respect to the scale shown in the following Table.

Table 1: Scale for pair wise comparison

Comparative Importance	Definition	Explanation
1	Equally important	Two decision elements (e.g., indicators) equally influence the parent decision element.
3	Moderately more important	One decision element is moderately more influential than the other.
5	Strongly more important	One decision element has stronger influence than the other.
7	Very strongly more important	One decision element has significantly more influence over the other.
9	Extremely more important	The difference between influences of the two decision elements is extremely significant.
2, 4, 6, 8	Intermediate judgment values	Judgment values between equally, moderately, strongly, very strongly, and extremely.
Reciprocals		If v is the judgment value when i is compared to j, then 1/v is the judgment value when j is compared to i.

- **Computing the Relative Weights**

AHP computes a weight for each decision element based on the pair-wise comparisons using mathematical techniques such as Eigenvalue, Mean Transformation, or Row Geometric Mean. We employ the Eigenvalue technique for computing the weights under AHP.

4. Implementation and the results

For implementing this algorithm we did the following:
4.1 Choose (8) images randomly as covers fig (1), all with the same size fig (2).
4.2 Choose (2) images as secret data (embedded image) fig (1), both with the same size fig (2).
4.3 Determine the features for all images (covers, and embedded).

Fig. 1: The covers and secret images used in experiment

cover and embedded image header information

	COVER	EMBEDDED
TOTAL FILE SIZE ...	921654	63994
PICTURE FILE OFFSET...	54	54
PICTURE WIDTH ...	640	156
PICTURE HIGHT ...	480	139
NO.OF BITS PER PIXEL...	24	24
total picture size ...	921600	63940
total picture color ...	16777216	16777216

Fig. 2: cover and embedded images specification

4.4 Features are organized according to priorities which are suggested by the user, for this work we suggested the following priorities:

a. ES (Estimated Secrecy).
b. R (Robustness).
c. ENT (Entropy).
d. CC (Capacity).
e. VS (Variance).
f. Ds (Energy).
g. CM (Histogram).
h. MS (Mean).

4.5 Determine the weight for each feature by using AHP process, as following:

Y3 =POWER((D3*F3*H3*J3*L3*N3*P3*R3);(1/8))

	C	D	F	H	J	L	N	P	R	Y	Z	AA
1	1	2	3	4	5	6	7	8	9	10	11	12
2	Parameters	CM	CC	VS	MS	ENT	DS	SEC	R	Eigenvalue	Priority Vector	weight
3	CM	1.000	0.250	0.333	2.000	0.200	0.500	0.125	0.142	0.362	0.031	3.12
4	CC	4.000	1.000	2.000	5.000	0.500	3.000	0.250	0.333	1.223	0.105	10.52
5	VS	3.000	0.500	1.000	4.000	0.333	2.000	0.200	0.250	0.818	0.070	7.03
6	MS	0.500	0.200	0.250	1.000	0.200	0.500	0.111	0.125	0.277	0.024	2.38
7	ENT	5.000	2.000	3.000	5.000	1.000	4.000	0.333	0.500	1.778	0.153	15.30
8	DS	2.000	0.333	0.500	2.000	0.250	1.000	0.166	0.200	0.522	0.045	4.49
9	SEC	8.000	4.000	5.000	9.000	3.000	6.000	1.000	2.000	3.884	0.334	33.42
10	R	7.000	3.000	4.000	8.000	2.000	5.000	0.500	1.000	2.759	0.237	23.74
11										0.000		
12										0.000		
13										0.000		
14	sum	30.500	11.283	16.083	36.000	7.483	22.000	2.685	4.550	11.623	1.000	100.00
15												
16												
17			λ=	8.23685	CI=	0.034	CR=	0.024				

Fig. 3: Priorities and weight of features

a. The value in each field in fig (3) for any row is calculated by comparing feature (parameters) in the row with each feature in the columns one by one, two at each time, and assigned value according to suggested priorities in section 4.4, and table (1).

b. Determine the Eigenvalue = ($\prod$ features values in each row)$^{1/n}$

where (n) is number of features in row.

c. Determine the priority vector where,

Priority for feature [i] = (Eigenvalue for feature [i]) / $\sum_{i=1}^{n}$ *Eigenvalue [i]*

d. Weight of feature [i] = priority [i] × 100

e. Inconsistent matrices typically have more than 1 eigenvalue. To check the consistency of the judgments, we have to measure the consistency ratio which should be less than one.

f. $\lambda_{max} = \sum_{i=1}^{n} sum_i \times priority_i$

g. CI (consistency index) =

$(\lambda_{max} - n) / (n-1)$

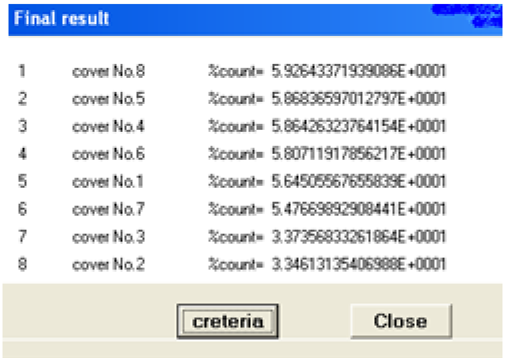
h. CR (consistency ratio) = CI / RI (**should be < 1**)

Random Consistency Index (RI) is obtained from Table 2 [12].

Table 2: consistency index

n	RI		n	RI
1	0		**6**	1.25
2	0		**7**	1.35
3	0.52		**8**	1.4
4	0.89		**9**	1.45
5	1.11		**10**	1.49

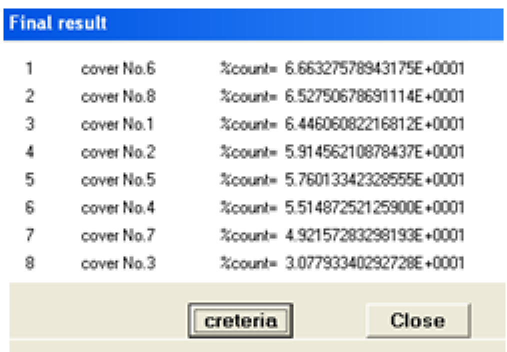
4.6 The final weight for each cover (when embedded images (1 and 2)) determined according to features weight calculated in AHP above where:
Final weight = CC+ENT+MS+VS +CM+DS+R+ES
The final results sorted in descending order, where the highest weight represents the best cover for embedding the specific image as shown in fig. 4.



Final result

1	cover No.8	%count= 5.92643371939086E+0001
2	cover No.5	%count= 5.86836597012797E+0001
3	cover No.4	%count= 5.86426323764154E+0001
4	cover No.6	%count= 5.80711917856217E+0001
5	cover No.1	%count= 5.64505567655839E+0001
6	cover No.7	%count= 5.47669892908441E+0001
7	cover No.3	%count= 3.37356833261864E+0001
8	cover No.2	%count= 3.34613135406988E+0001

a) Result when use embedded 1



Final result

1	cover No.6	%count= 6.66327578943175E+0001
2	cover No.8	%count= 6.52750678691114E+0001
3	cover No.1	%count= 6.44606082216812E+0001
4	cover No.2	%count= 5.91456210878437E+0001
5	cover No.5	%count= 5.76013342328555E+0001
6	cover No.4	%count= 5.51487252125900E+0001
7	cover No.7	%count= 4.92157283298193E+0001
8	cover No.3	%count= 3.07793340292728E+0001

b) result when use embedded 2

Fig 4: Final weight when calculate features with both embedded 1 and embedded 2

5. Prove the Results

Perfect steganography is when we get stego-image similar to original cover by both perceptual and computer reading. This may be impossible to reach.
In our work we hope to choose cover, give the closest features to original cover when it changes to stego-image.
To prove this we try to apply the following step, which helps us to evaluate our work

5.1 First convert each cover to stego-object (by hiding each embedded image in all covers) by using (LSB) hiding technique.

5.2 Determine the perceptual difference between the origin cover and stego image fig (5).

5.3 Determine the histogram for origin image and stego image fig. (6).

5.4 Determine the similarity between the cover image and the corresponding stego-object. Formally, similarity can be defined via similarity function [3].
Let c be nonempty set.
Function Sim: $c^2 > [-\propto, 1]$ is called similarity function on c,
if for $(x, y) \in c$ *Sim (x, y) =1* iff $x = y$
For $x \neq y$, sim $(x, y) < 1$
Perfect similarity ≈ 1
In the case of digital images the correlation between two images can be used as similarity function. Therefore most practical steganographic systems try to fulfill the condition

Sim (cover, stego) =1

Similarity determine by comparing both of cover and stego image.

5.5 Determine the security for stego-object by using Eq. (11).

Perfect security = **0.**

5.6 Determine the PSNR.

Fig. 5: comparing cover image before and after hiding embedded 1

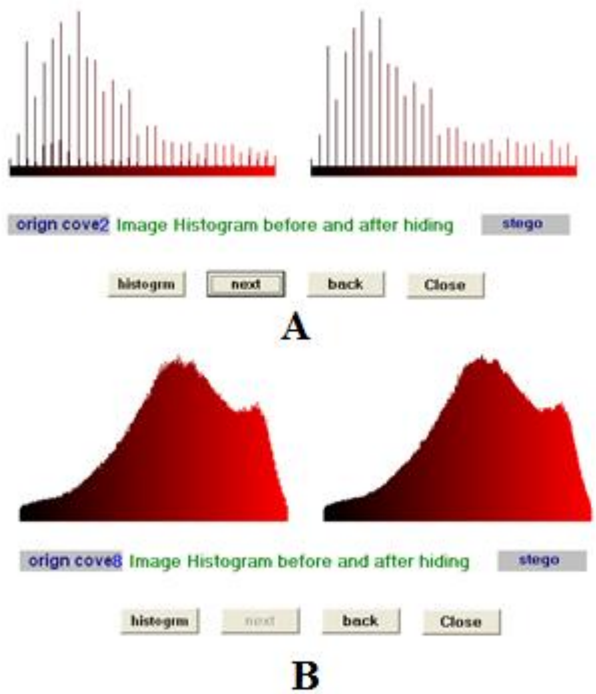


Fig 6: Histogram for both origin and stego images for covers (1 and 8) when hiding embedded image 1

Table 3: Comparing result when hiding embedded 1 in covers.

cover	similarity	secrecy	PSNR
1	0.90610	0.00803	34.310
2	0.90143	0.15486	34.239
3	0.90303	0.15538	34.262
4	0.90604	0.00708	34.311
5	0.90633	0.00800	34.311
6	0.90607	0.00778	34.317
7	0.90555	0.01082	34.300
8	0.90687	0.00203	34.318

Table 4: Comparing result when hiding embedded 2 in covers.

cover	similarity	secrecy	PSNR
1	0.86755	0.64115	33.812
2	0.86750	0.64115	33.816
3	0.86120	0.64118	33.745
4	0.86054	0.64114	33.738
5	0.86835	0.64026	33.821
6	0.86793	0.62098	33.832
7	0.86349	0.63030	33.769
8	0.86551	0.64118	33.766

It is clear from the results above the following

A. There is no perceptual difference between origin and stego image.

B. Histogram of origin and stego image is almost the same.

C. The values of (similarity, security, and PSNR) confirm the result in fig. 4 for both cover 8 when embedding embedded image1 in it, and cover 6 when embedded the embedded image 2 in it. Almost both of them give the best result.

6. Conclusions

This paper introduced a novel algorithm to choose cover from many suggested covers; it is the first algorithm discusses this problem.
The algorithm proved by using LSB image in image steganography, and measuring the perceptual and computer reading similarity, PSNR, security, and histogram to prove the efficiency of the algorithm.
Tables (3, 4) proved the results in fig. (4) and the best cover in fig. (4) get the best result when comparing stego-image with the cover images, at the same time the cover with the minimum weight gets worst result in comparing stego-image with cover image.
AHP algorithm used to count the weight of each feature. Final results may change if the features priorities will be changed, due to change of weight.
From all the results, we can say, that we proposed and built dependable algorithm, and by using other images features, we can develop this algorithm to become more accurate.
We suggest for future works, determine the features for each channel of the image color (Red, Green, and Blue).

References

[1] Abbas Cheddad, JoanCondell, KevinCurran, PaulMcKevitt, "Digital image steganography: Survey and analysis of current methods", Journal signal processing, Volume 90, Issue 3, 2010

[2] Gerhard X. Ritter; Joseph N. Wilson, Handbook of Computer Vision Algorithms in Image Algebra, CRC Press LLC , 1996

[3] Gonzalez R.C. and Woods R.E, Digital Image Processing, 3rd edition , Prentice Hall, 2008.

[4] Hedieh Sajedi, M. Jamzad ,"Contourlet-Based Steganography Using Cover Selection", International Journal of Information Security, Springer, vol. 9, no.5, 2010, pp. 337-345.

[5] İsmail Avcıbaş "Image Quality Statistics and their use in Steganalysis and Compression", PhD thesis, Boğaziçi University, Istanbul, Turkey, 2001

[6] Katznbeisser S., Petitcolas F., Information Hiding Teqniques for Steganography and Digital Watermarking, artech house, 2000.

[7] K B Shiva Kumar , K B Raja, R K Chhotaray, Sabyasachi Pattnaik, "Steganography Based on Payload Transformation", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011

[8] Maurice Maes " Twin Peaks: The Histogram Attack to Fixed Depth Image Watermarks", Information hiding second international workshop proceedings, 1998, vol. 1525 of lecture notes in computer science Springer pp 290-305

[9] Mehdi Kharrazi a, Husrev T Sencar b, Nasir Memon, "Cover Selection for Steganographic Embeddeing", IEEE International Conference on Image Processing, 2006.

[10] Ross J. Anderson "Stretching the Limits of Steganography" , proceedings of the first international workshop on information hiding, 1996, springer-Verlag, London, UK, pages 39-48.

[11] Saaty, Thomas. L., Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process, RWS Publications, Pittsburgh, PA. 1996

[12] **Saaty, Thomas L**. "Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors - The Analytic Hierarchy/Network Process". *RACSAM (Review of the Royal Spanish Academy of Sciences, Series A, Mathematics)* **102** (2): 251–318., 2008

[13] V. Lokeswara Reddy, A. Subramanyam, P. Chenna Reddy, "Implementation of LSB Steganography and its Evaluation for Various File Formats", Int. J. Advanced Networking and Applications, Volume: 02, Issue: 05, Pages: 868-872, 2011.

[14] Weiming Zhang, Shuozhong Wang, and Xinpeng Zhang," Improving Embedding Efficiency of Covering Codes for Applications in Steganography", IEEE communicatons letters, vol. 11, no. 8, 2007

[15] Westfeld, A. "High Capacity Despite Better Steganalysis: F5 – a Steganographic Algorithm." *Proceedings of the 4th Information Hiding Workshop, Lecture Notes in Computer Science 2137*, 2001, pages 301-314.

[16] Yong Xu, Hui Ji, "Viewpoint Invariant Texture Description Using Fractal Analysis", International Journal of Computer vision, Volume 83, Issue 1, pages 85-100, 2009

[17] Zöllner, J., H. Federrath, etl. "Modeling the Security of Steganographic Systems." Proceedings of the 2nd Workshop on Information Hiding, Lecture Notes In Computer Science. 1998, Springer-Verlag, pages 344-354.

Nidhal El-Abbadi received BSc in chemical engineering, BSc, MSc, and PhD in computer science, worked in industry and many universities, he is general secretary of colleges of computing and informatics society in Iraq, Member of Editorial Board of Journal of Computing and Applications, reviewer for a number of international journals, has many published papers and three published books (programming with Pascal, C++ from beginning to OOP, Data structures in simple language), his research interests are in image processing, biomedical, and steganography, He's Associate Professor in Computer Science in the University of Kufa – Najaf, IRAQ.