A High Quality Image Steganography Scheme Based on Fuzzy Inference System

Sara sajasi
Department of Electrical, Computer and Biomedical Engineering
Islamic Azad University, Qazvin branch
Qazvin, Iran
s.sajasi@qiau.ac.ir

Amir Masoud Eftekhar Moghadam
Department of Electrical, Computer and Biomedical Engineering
Islamic Azad University, Qazvin branch
Qazvin, Iran
eftekhar@qiau.ac.ir

Abstract—This paper presents a novel image steganography approach for hiding a secret image in the cover image. It benefits from the fact that the occurrence of some local features in the image demonstrates remarkable patterns for determining the payload of the cover image sub-regions based on Human Visual System (HVS). However, the goal of this paper is to propose a system deals with improving the visual quality of the stego-image, while still providing a large embedding capacity. To achieve this goal, a Fuzzy Inference System (FIS) is designed as a classifier which adopts the local features of the cover image sub-regions as its crisp input values and produces semantic concepts corresponding to the payload of image sub-regions. This ensures to decrease the rate of changes in the stego-image, even after embedding a large amount of secret bits. The experimental results demonstrate the effectiveness of the proposed approach.

Keywords—image steganography; Fuzzy Inference System (FIS); Human Visual System (HVS).

I. INTRODUCTION

Nowadays, information security is playing a major role in multimedia communication due to the rapid growth of transmitting data through public channels such as internet. Steganography is an effective technique for security systems carried out by embedding important data in multimedia such as images, audios or videos.

Many steganography schemes for digital images have been proposed [1]. One of the most commonly used techniques is Least-Significant-Bit (LSB) steganography in which only the LSB planes of the cover image are directly replaced with the secret bits [2]. Despite the simplicity of the LSB steganography scheme, it has been shown in [3] this technique is not secure against statistical attacks. It is because of the embedding payload which is the same for all of the pixels of an image. To solve it, several adaptive methods have been proposed in which the payload of each pixel is variable [4]-[6]. Based on sensitivity of human vision in smooth areas, Pixel-Value Differencing” (PVD) techniques have been proposed in [4]-[5] where the size of payload can be determined adaptively according to the different value between neighboring pixels. In [6], besides employing LSB substitution technique, a hybrid edge detector method has been considered to compute the actual edge pixels in an image to conceal more secret bits in.

Recently, intelligent algorithms based on soft computing, such as Fuzzy Logic (FL), Adaptive Neural Networks (ANNs), Genetic Algorithms (GA), and Particle Swarm Optimizer (PSO), were used in steganography, to achieve robust, optimal and adaptive solutions, simultaneously. [7] proposed a secure steganography method against the RS analysis by modifying the pixel values of the stego-image using GA. PSO was developed in [8] to improve the quality of the stego-image by deriving an optimal substitution matrix for transforming the secret messages. An efficient concept of developing a high payload steganographic approach is presented in [9] using the hybrid ANNs and modified Adaptive GA employing Uniform Adaptive Relaxation (ANN-AGAUAR).

The major contribution of this paper focuses on developing a novel adaptive steganography scheme for hiding a secret image in the gray-scale images which can increase the visual quality of the stego-image with providing an acceptable embedding payload. The advantage and strength of the proposed scheme is obtained by an intelligent technique integrating Fuzzy Inference System (FIS) with the characteristics of Human Visual System (HVS). Due to the fact that changes in complex areas of an image are more difficult to be detected, first three local features, including texture, edge, and brightness are extracted from the each block of the image. Then, the extracted features are feed as the input values to the FIS to determine the payload of each image block as the output, adaptively. All blocks are classified into five different types in which more secret bits are embedded into the pixel values located into the complex blocks. Finally, the LSB substitution approach is employed to hide the secret image into the cover image pixels.

The rest of this paper is organized as follows. In section 2, the Fuzzy edge detector is discussed. The details of our new algorithm will be presented in section 3. Section 4 covers the experimental results and discussions. Finally, the whole paper is concluded in section 5.

II. FUZZY EDGE DETECTOR

Edge detection is an important topic in computer vision and image processing which is used to analyze the local properties of an image. The edge pixels indicate the boundary between two regions in an image. In this paper, we use a...
simple and precise fuzzy complement edge detector [6] due to the flexibility in dealing with the ambiguity and vagueness of the edge pixels. The details of the fuzzy edge detector is briefly presented in three steps as follows:

Let us \( C \) be the image, of size \( M \times N \), and all the pixels are gray level from 0 to 255, i.e. \( c_{mn} \in [0, 255] \) with \( m, n \in [1, M] \). First, the membership grade value \( \mu_{mn} \) at position \((m, n)\) should be determined. The image \( C \) is transformed into an array \( F \) of fuzzy singletons, \( \mu_{mn} \). \( F \) is an array which is a union of all \( \mu_{mn} \)'s and is determined as follows:

\[
F = \bigcup_{m=1}^{M} \bigcup_{n=1}^{N} \mu_{mn}, \quad \mu_{mn} = \frac{c_{mn}}{c_{\text{max}}} \tag{1}
\]

Where \( c_{\text{max}} \) is the biggest gray-scale value in image \( C \). Then, the degree of edginess, \( \mu_{mn} \), for each pixel \( c_{mn} \) with a \( k \times k \) sliding window is determined as follows:

\[
\mu_{mn} = \min\left(1, \frac{\tau}{k} \sum_{i} \sum_{j} \min\left(\mu_{ij}, 1 - \mu_{ij}\right)\right)^{1/\rho} \tag{2}
\]

Where the value of ‘\( \tau \)’ and ‘\( k \)’ is suggested to be 9 and 3, respectively. Also, \( \mu_{mn} \) is locally calculated inside each window \( K \) as follows:

\[
\mu_{mn} = \frac{\max(c_{ij}) - \min(c_{ij})}{c_{\text{max}}}, \quad i, j \in [1, w] \tag{3}
\]

With a predefined threshold value, \( th \in [0, 1] \), the pixel \( c_{mn} \) is classified into the edge pixel and non-edge pixel as (4):

\[
c_{mn} \text{ is } \begin{cases} \text{edge pixel} & \text{if } \mu_{mn} \geq th \\ \text{non-edge pixel} & \text{otherwise} \end{cases} \tag{4}
\]

Finally, the edge image, \( F' \), is determined as (5):

\[
F' = \bigcup_{m=1}^{M} \bigcup_{n=1}^{N} c_{mn} \tag{5}
\]

Fig. 1 shows the edge image which is generated from this strategy for the cover image, “Lena”, of size 512×512. It can be seen that the edge pixel can clearly and precisely be detected.

III. THE PROPOSED SCHEME

In this section, a new steganography scheme is thoroughly investigated. The proposed scheme consists of two processes: adaptive embedding process, and adaptive extracting process.

A. Adaptive embedding process

In this section, the main goal is to hide a secret image into a cover image with high imperceptibility. Adaptive embedding process, as shown in Fig. 2, is detailed in the following three sections: Feature extraction, Fuzzy Inference System (FIS), and embedding capacity estimation.

1) Feature extraction

Suppose \( S \) is the gray-scale secret image and \( C \) is the gray-scale cover image for extracting features. First, \( C \) is partitioned into non-overlapping 3×3 blocks. Each block contains 9 pixels as \( p_i, 0 \leq i \leq 8 \). Then, three local features are extracted from each block individually as follows:

1.1) Texture feature (\( T_k \))

Embedding more secret bits in the stronger textures is less visible by HVS. The texture, \( T_k \), can be estimated based on a statistical measurement, entropy which is used to describe how much information exists in the system. For each block, entropy [10] is defined as (6) where \( p \) contains the histogram counts:

\[
T_k = \text{entropy} = -\sum(p \times \log_2(p)) \tag{6}
\]

1.2) Edge sensitivity (\( E_k \))

Existing many edge pixels in a block indicates that the block belongs to the complex one in which more secret bits can be embedded. In order to extract the edge pixels, a fuzzy edge detector (introduced in section II) is employed on \( C \). The states of each pixel, \( p_i \), in the edge image, \( C' \), is defined as ‘1’ if \( p_i \) is an edge pixel, otherwise, it is defined as ‘0’. Then, for each block of \( C' \), the mean value of its pixels, \( E_m \), is defined as (7) to determine the complexity of each block.

\[
E_m = \frac{\sum p_i'}{9}, \quad 0 \leq i \leq 8 \tag{7}
\]

1.3) Brightness sensitivity (\( B_k \))

Human visual system is less sensitive toward brighter regions than darker ones. For gray-scale images, each pixel has a 8-bit format with a value between ‘0’ to ‘255’. The value of ‘0’ indicates that the pixel has maximum darkness, while the value of ‘255’ represents the maximum brightness. Other values in between represent the gray shades. In order to
estimate how much bright is a block, the mean of pixel value, \( B_m \), is calculated in it as follows:

\[
B_m = \frac{\sum_{i} p_i}{9}, \quad 0 \leq i \leq 8 \quad (8)
\]

2) **Fuzzy Inference System (FIS)**

In this section, we describe the process of designing the Fuzzy Inference System (FIS) as a new approach to determine the complexity of each block in image steganography.

2.1) **Fuzzifier**

A fuzzifier performs the function of fuzzification which converts the crisp input to fuzzy values through MFs of features. In this paper, crisp inputs are the values obtained from all three local features presented in previous section. As it shown in Fig. 3a-c, we consider four MFs for \( T_k \), three MFs for \( E_k \), and three MFs for \( B_k \) as the inputs of the fuzzy system. Also, Fig. 3d shows MFs of five semantic classes as the output of the fuzzy system, \( W \).

2.2) **Fuzzy inference engine**

In fuzzy systems, the inference engine utilizes the information rules to assign semantic concepts to input feature vectors. The inference engine which is used in the proposed scheme is a Mamdani type [11]. In the proposed scheme, we consider 36 rules in total which are defined in such a way to discover strong relationships between input features and output classes. Some samples of the 36 rule is defined as follows.

\[ R_1: \text{if } T_k \text{ is Smooth} \land B_k \text{ is dark} \land E_k \text{ is Low then } W \text{ is Very Small.} \]

\[ R_2: \text{if } T_k \text{ is Smoothly Smooth} \land B_k \text{ is dark} \land E_k \text{ is Low then } W \text{ is Very Small.} \]

\[ R_3: \text{if } T_k \text{ is Smoothly Smooth} \land B_k \text{ is dark} \land E_k \text{ is High then } W \text{ is Medium.} \]

\[ R_4: \text{if } T_k \text{ is Smooth} \land B_k \text{ is Dim} \land E_k \text{ is Low then } W \text{ is Very Small.} \]

\[ R_5: \text{if } T_k \text{ is Smoothly Smooth} \land B_k \text{ is Dim} \land E_k \text{ is Medium then } W \text{ is Small.} \]

\[ R_6: \text{if } T_k \text{ is Smoothly Smooth} \land B_k \text{ is Dim} \land E_k \text{ is High then } W \text{ is Medium.} \]

\[ R_7: \text{if } T_k \text{ is Smoothly Rough} \land B_k \text{ is Dim} \land E_k \text{ is Medium then } W \text{ is Medium.} \]

\[ R_8: \text{if } T_k \text{ is Smoothly Rough} \land B_k \text{ is Dim} \land E_k \text{ is High then } W \text{ is Large.} \]

\[ R_9: \text{if } T_k \text{ is Smoothly Rough} \land B_k \text{ is Bright} \land E_k \text{ is Low then } W \text{ is Small.} \]

\[ R_{10}: \text{if } T_k \text{ is Rough} \land B_k \text{ is Bright} \land E_k \text{ is High then } W \text{ is Very Large.} \]

2.3) **Defuzzifier**

In order to apply the fuzzy output values directly as such to the application, it is necessary to convert the fuzzy output into a crisp value. This process is always performed via a defuzzifier. In this paper, after a lot of analysis, we employ the Mean of Maxima (MOM) defuzzifier due to the better...
Accuracy on results. The system output, $d$, would be semantic concepts corresponding to the block types in the cover image.

3) Embedding capacity estimation

The number of bits should be embedded in each block is determined adaptively, according to the value of $d$. All blocks are classified into five different types:

- **Block type I**: If $2.0 \leq d$, then the block belongs to type I, which is located in a rather smooth area. In this case, each pixel of Block of type I can hold just 1 secret bit.
- **Block type II**: If $4.0 < 2.0 \leq d$, then the block belongs to the type II where the payload is just 2 bits for each pixel.
- **Block type III**: If $6.0 < 4.0 \leq d$, then the block belongs to the type III where the payload of each pixel is 3 bits.
- **Block type IV**: If $8.0 < 6.0 < d$, then the block belongs to type IV where the payload of each pixel is 4 bits.
- **Block type V**: If $d > 8.0$, then the block belongs to the type V which is located in a very complex area. In this case, each pixel of Block type V can hold 5 secret bits.

In our proposed scheme, each block needs to spend 2 bits, identifier bits, to determine the block type. For this purpose, we use ‘00’ to identify the Block type I and IV, ‘01’ to identify the Block type II and V, and ‘10’ to identify the Block type III due to the fact that boundaries of the neighboring blocks may have overlap with each other. Finally, using LSB substitution method, the stego-image, $C'$, will be generated.

![Fig. 3. Membership functions and semantic classes; (a) input feature $T_k$, (b) input feature $E_k$, (c) input feature $B_k$, (d) output semantic classes.](image)

![Fig. 4. (a) The cover images; (1) Baboon, (2) Houses, (3) Man, (4) Lena, (b) The secret image; Girl.](image)
B. Adaptive extracting process

The extracting process is the inverse of the embedding process. To extract the secret image, $S$, first three local features, Texture, Edge, and Brightness are calculated for each $3 \times 3$ block of the image, $C'$, as explained in section III-A-1. Second, by running FIS on these features, we obtain the output of defuzzification, $d$, for each image block. Then, according to $d$ and the identifier bits, the type of each block can be identified as follows:

1. If $d \leq 0.4$ :
   (a) If the identifier bits are '00', then this is a Block type I.
   (b) If the identifier bits are '01', then this is a Block type II.
   (c) If the identifier bits are '10', then this is a Block type III.

2. If $0.4 < a \leq 0.6$ :
   (d) If the identifier bits are '01', then this is a Block type II.
   (e) If the identifier bits are '10', then this is a Block type III.
   (f) If the identifier bits are '00', then this is a Block type IV.

3. If $d > 0.6$ :
   (g) If the identifier bits are '00', then this is a Block type IV.
   (h) If the identifier bits are '01', then this is a Block type V.
   (i) If the identifier bits are '10', then this is a Block type III.

IV. EXPERIMENTAL RESULTS

To conduct our experiments, the four well-known images as 8-bit grayscale of size $512 \times 512$ are used as the cover images as shown in Fig. 4a. We also use a secret image, “Girl”, as shown in Fig. 4b, with different sizes; $128 \times 128$, $256 \times 128$, $256 \times 256$, and $384 \times 256$. The experiments are grouped in two categories. First, the stego-image quality and embedding capacity are investigated. Second, the proposed scheme is compared with other steganographic methods.

A. Evaluation of stego-image quality against embedding payload

The embedding capacity is calculated by the number of secret bits concealed in each cover image. The quality is evaluated by subjective and objective measurements.

1) Objective measurement

PSNR is an objective measurement which is defined in (9):

$$\text{PSNR}(C, C^*) = 10 \times \log_{10} \left( \frac{M \times N \times (255)^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} (C(i,j) - C^*(i,j))^2} \right)$$

Where $C(i,j)$ and $C^*(i,j)$ represent the pixel values of the cover image and the stego-image located at $(i, j)$, respectively. $M$ and $N$ refer to the width and height of the cover image, respectively.

Fig. 5 exhibits the results of embedding capacity and PSNR value, when embedding different sizes of Girl in cover images. The overall PSNR of each image is different according to the complexity of the cover image. Indeed, lower complexity
of the cover image results in higher PSNR. “Lena”, for example, is a smooth image which tends to have more smooth blocks where the payload of each pixel is 1 or 2 bits. In contrast, “Baboon” is a more complex image which has rich blocks where the payload of each pixel is 1 or 2 bits. In this reason, the PSNR is decreased but the visual quality of the stego-image will be better than smoother images. Because complex blocks have more tolerance against changes produced after embedding the secret bits in it. Moreover, complex images have larger payloads than smoother images because of having more complex blocks. As seen in Fig. 5, by increasing the size of secret image, the PSNR value is still larger than 46dB for all cover images. It indicates that we achieve a high quality stego-image with an acceptable embedding capacity.

2) Subjective measurement

HVS is a subjective measurement which referred to examining the stego-image with naked eye to identify any obvious distortion. Therefore, we employ HVS metric to evaluate the quality of the stego-image, since PSNR can not be a precise measurement, individually. Fig. 6 shows the quality of the stego-images that are generated by the proposed method for the test image Lena. From the HVS, it can be seen that stego-images are visually indistinguishable from the corresponding cover image even when 5 LSBs of each pixel in Block type V are replaced with secret image bits. It is due to the fact that changes in edge areas do not seriously affect the stego-image quality as seen by HVS.

B. Comparative analysis of the proposed scheme with other schemes

To present the superiority of the proposed scheme over the previous schemes, we have compared it with two related schemes in a same condition; the classic LSB substitution technique and the edge-based steganography approach proposed in [6]. Table I shows the performance comparisons of the proposed scheme and the two other related schemes. The comparison is based on PSNR (dB) to determine the visual quality after embedding four sizes of the secret image. Compared to previous works, the proposed scheme can achieve the greater PSNR value, especially in higher payloads, while the embedding payload is the same. In fact, our proposed method can preserve more image details and avoid serious visual quality degradation in higher payloads.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena (512×512)</td>
<td>51.15</td>
<td>51.14</td>
<td>50.64</td>
</tr>
<tr>
<td>Baboon (512×512)</td>
<td>49.64</td>
<td>49.77</td>
<td>49.91</td>
</tr>
</tbody>
</table>

V. Conclusion

In this paper, an adaptive steganographic scheme is proposed based on Fuzzy Inference System (FIS) and Human Visual System (HVS). We extract three local image features from the image blocks, which can well reflect the visual characteristic of the image. Then, by running FIS on these three features, an output is obtained which is used to determine the embedding payload of each block pixels adaptively. The proposed approach can simultaneously improve visual quality and embedding capacity of the stego-images. Experimental results demonstrate that compared with other steganographic methods, the new scheme produces higher quality stego-images and payloads.

REFERENCES