IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 12, 2019, accepted January 17, 2020, date of publication January 20, 2020, date of current version February 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968217

An Accurate Texture Complexity Estimation for Quality-Enhanced and Secure Image Steganography

AYESHA SAEED[1], FAWAD[1], (Student Member, IEEE), MUHAMMAD JAMIL KHAN[1], (Member, IEEE), HUMAYUN SHAHID[1], (Member, IEEE), SYEDA IFFAT NAQVI[1], MUHAMMAD ALI RIAZ[1], (Member, IEEE), MANSOOR SHAUKAT KHAN[2], AND YASAR AMIN[1], (Senior Member, IEEE)

[1]ACTSENA Research Group, Telecommunication Engineering Department, University of Engineering and Technology Taxila, Taxila 47050, Pakistan
[2]Mathematics Department, COMSATS University Islamabad, Islamabad 45550, Pakistan

Corresponding author: Fawad (engr.fawad@students.uettaxila.edu.pk)

This work was supported in part by the Higher Education Commission (HEC) of Pakistan through Technology Development Fund under Grant TDF-67/2017, and in part by the ASR&TD-UETT Faculty Research Grant.

ABSTRACT Content-adaptive steganography intends to hide data in the complex texture content of the image. Recently, some secure steganography methods have been proposed to identify the textural complexity of an image. However, most of the techniques do not take into account the information of pixel variation around the central pixel in all possible directions and therefore they are unable to accurately analyse the texture complexity. This work offers a quality-enhanced and secure method of content-adaptive image steganography. The proposed method is divided into three sequential steps: image segmentation, pixel complexity identification, and data embedding. An input cover image is initially divided into small local regions and the pixel-complexity is identified based on the proposed Complex Block Prior (CBP) criterion. In a local block, a high pass filter (HPF) bank is applied and eight residual responses are obtained. Following the CBP criterion, a complexity level out of nine levels is assigned to an individualized pixel block. The pixels are then arranged in the priority of complexity from highest to lowest. Data embedding for the corresponding complexity level then takes place using the proposed adaptive embedding algorithm. Experimental results verify the preservation of visual quality of stego images produced by the proposed method. Three image datasets: Standard test images, BOWS2 and BOSS-base are used for the experimentation and comparison with prior state-of-art methods. Highest values of the IQ (image quality) parameters e.g., SSIM and WPSNR show the effectiveness of the proposed method.

INDEX TERMS Noisy texture, content adaptive, pixel selection, data embedding, complex block, complexiy estimation.

I. INTRODUCTION

With the advancement in networking technology, the digitization and use of high-speed communication links have given rise to immense possibilities [1]–[3]. As a result, the data communicated over the internet is increasing day by day. The communication over insecure network links is vulnerable to attacks from eavesdroppers. These attacks include illegally copying, modifying and misusing the information in felonious activities [4]–[7]. Consider the following application areas: Password transmission in a client-server environment, personal document sharing, biometric data transfer, medical record storage, bank account details storage, social media content sharing, TV broadcast, storage of data in cloud platforms, etc. With the aim of confidential communication over an insecure network, significant amount of efforts has been made under the field of security systems [8], [9]. A generic security system aims to achieve two goals: information access control and data integrity [10]. The classification of the field of security systems is as follows:

The associate editor coordinating the review of this manuscript and approving it for publication was Yuming Fang.

1) Information encryption and 2) Information hiding. Cryptography is an information encryption technique and is the process of altering secret data such that the data is not meaningful [11]. There is no separate cover medium to carry the secret data. Although the information is uninterpretable, this unintelligent representation raises suspicion and therefore is prone to cryptanalysis attacks. In contrast, information hiding techniques such as watermarking and steganography uses a cover medium to embed secret data and therefore, the existence of the message is hidden behind the cover. In the context of this work, information hiding has one of two definitions. It is defined as: ''imperceptibly embedding in a cover'' e.g., watermarking or ''making the existence of data secret'' e.g., steganography. Watermarking is defined as ''the process of altering a cover medium to embed message about the cover''. The aim of a watermark is to protect the copyright (ownership) information of the marked signal and may be made visible to claim the ownership. Steganography, on the other hand, is defined as ''the process of altering a cover medium to embed a secret message''. The aim of steganography is to keep the existence of message secret other than the intended person. Therefore, the objective behind the embedding of secret data differentiates watermarking and steganography. A generic image steganographic system is shown in Fig. 1. It consists of following basic components.

1) Medium/carrier
2) Secret message
3) Embedding algorithm
4) Steganographic key
5) Extraction algorithm

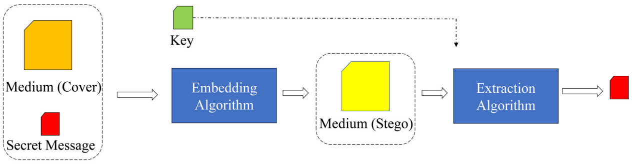


FIGURE 1. A generic image steganographic system representing the basic components of image steganography.

In digital steganography, a digital object/medium is used for hiding data e.g. text, audio, image, video and data packets, etc. A good carrier signal must have two necessary qualities. 1) presence and usage in abundance so that it is difficult to select the signal for detection of message. And 2) redundancy in data representation. Currently, the most secure and efficient medium of steganography is the image. Images are widely communicated over the internet and have the psychovisual redundancy [12] which is used to conceal the secret information without losing visual quality. A steganographic algorithm is used to embed secret payload in the digital object. The object before embedding is known as cover and after embedding the object is called stego. At the extraction side, a known key is used by a person which specifies the locations in which data is hidden. The key is either hidden inside the image or is shared through some other means. Image steganography can be carried out in both spatial and transform domain [13], [14]. Spatial domain steganographic algorithm directly embeds in the pixels of the cover image. Transform domain steganography consists of transforming the spatial representation of an image in frequency domain and then using the coefficients for data embedding. Spatial domain steganography provides more capacity of embedding while transform domain techniques perform better in security analysis. The performance of an embedding algorithm is analyzed based on the following three factors: payload capacity, visual quality, and undetectability.

Embedding payload is the maximum capacity that can be embedded in an image. An image's visual quality is defined as the perceived distortion or visual artifacts which are generated when the image pixels are modified. Generally, the more the embedding capacity the more is the visual distortion introduced in the image and the poor is the visual quality. Another parameter that plays a significant role in determining a steganographic algorithm's performance is the undetectability of a stego image in steganalysis domain. Steganalysis is the counterpart of steganography. It tries to detect the presence of hidden data in a given image. The most successful steganalysis in today's literature is carried through the statistical detection methods implemented using machine learning [15]. The state of the art steganalysis methods are based on the estimation of local pixels. It is easy for a detector to predict local pixels which lie in a smoothly varying region. Hence, the pixels in such areas if used for embedding are vulnerable. The highly varying regions of the image are safe for embedding since the detectors are unable to accurately predict the pixel value. Following the idea, the concept of content-adaptive steganography is developed which selectively embeds into high texture content of the image.

In order to understand the content-adaptive approach of image steganography, we first discuss a non-adaptive approach e.g., the simple LSB substitution (LSBS) method. A pseudorandom number generator selects the embeddable pixels equally from the smooth and textured regions for data embedding [16]. The LSB of chosen pixels is then replaced with the message bit. LSBS degrades the visual quality of the cover image since the embedded pixels in the smooth region become prominent and appear as a visual artifact. LSBS also introduces a structural asymmetry in the cover image. When the LSB of an even pixel is replaced with a message bit, the number of even and odd pixels in the cover image are unbalanced and the structure of the cover image is disturbed. The structural steganalysis attacks such as weighted stego, RS analysis and sample pair analysis exploit this structural asymmetry and easily detect the existence of distortion caused by data embedding [15]. The statistical steganalysis attacks (SPAM and SRM) on the other hand use the information of the smoothness of the image and can predict the pixels especially in the smooth areas of the image. The security of simple LSBS based methods can be improved when the structural asymmetry is reduced and embedding changes are restricted to the complex texture areas of the image. These two approaches when combined provides high visual quality as well as high security.

An edge-adaptive scheme based on LSB Matching Revisited (EA-LSBMR) is presented in [17]. LSBMR achieves reduced modification rate as compared to simple LSBS. LSBMR embeds two bits of data in a pixel pair in such a way that first data bit is embedded in the first pixel and next bit is embedded in even-odd relation of the pixel pair. The modification rate is reduced to 0.375 bits/pixel compared to 0.5 bits/pixel in LSBS. The edge adaptivity is introduced in the steganographic scheme. A 1x2 high pass filter (HPF) is utilized to measure the business of the pixel pair and a difference threshold is used to select only the busiest pixel pairs. The threshold is adaptively changed according to the size of secret data. An improved edge detection method is presented in [18]. The edge adaptive image steganography (EIS) scheme uses a canny edge detector to locate embedding pixel locations and adopts a 2-bit LSBS method for data embedding. High-security performance is achieved as compared to HUGO, EA-LSBMR, PVD, and HBC while comparable security is achieved compared to S-UNIWARD. Another improved edge detection method is presented in [19]. A modified median edge detector (MMED) is used to exploit the edges in two directions (horizontal and vertical). The highest edge value is used for further processing and thus only the sharper edge is chosen. The MMED operator is applied for every pixel and an MMED matrix is formed which is then divided into three groups based on the edge intensity. The threshold levels for the division of the edge intensity matrix is determined adaptively based on message length. The first group which contains the sharpest edges is used for 3bit LSBS while the second and third groups are used for 2 and 1-bit LSBS respectively. A pixel value difference (PVD) based technique in combination with LSBS is proposed in [20]. The scheme partitions an image into non-overlapped 3x3 blocks and uses the nine pixels in each block for data embedding. For each pixel in a block, six higher bit planes or the ''quotient value'' (in decimal terms) is utilized for PVD embedding while two lower bit planes are utilized for LSBS. In the case of PVD embedding, the difference of center pixel quotient value is evaluated in eight directions. The difference values are updated with the decimal value of a set of message bits and the number of message bits are determined based on pre-defined capacity values. The embedded differences are transformed to the corresponding stego quotients and a mean center quotient is obtained from the eight stego quotients. The neighboring quotients are also updated according to the mean center quotient. In parallel to PVD embedding, LSBS embedding is performed in the two lower bit planes. If the falloff boundary (FOBP) occurs as a result of data embedding then the whole embedding process is undone and a simple 4-bit LSBS embedding is performed on the lower bit planes. The presented scheme achieves a very high capacity of 4.5 bpp while resisting RS and PDH steganalysis.

The aim of this paper is to present a quality-enhanced and secure methodology of image steganography. This requires accurate identification of the local texture complexity. To analyze the texture, the underlying assumption is: the more the neighborhood pixels of the targeted region are included in the texture analysis, the accurately is the texture analyzed. Therefore, we propose a method of the computation of complexity of a local block for adaptive pixel-selection. Following are the contributions of the paper.

1) The presented algorithm achieves more embeddable pixels by partitioning the image into small blocks of variable size depending on the neighbors of the central pixel. Moreover, with the use of overlapping blocks the texture complexity estimation is now achieved for every pixel.
2) High capacity of embedding is achieved in terms of the multibit embedding and the quality enhanced result of steganographic algorithm.
3) We use an eight directional high pass filter bank to compute the eight difference values of the pixel block. The filter is designed to calculate the difference between the central pixel and each of its eight neighbors.
4) We define a novel complex block prior (CBP) criterion which defines nine complexity levels. Following the criterion, a threshold/difference range classifies the eight differences into two groups. Based on the number of differences in a group, a complexity level is assigned to the pixel block.
5) We use a method of combining the eight difference responses in the CBP criterion and use the single value to arrange the pixel blocks in the order of complexity from highest to lowest.
6) We derive an expression to combine eight difference values into a single value and use the value to estimate the number of bits to be embedded per pixel.
7) An adaptive setting is devised which selects the embedding algorithm based on single or multibit embedding.

Remaining contents of the paper are organized as follows. Sect. II reviews methodology of important related techniques, Sect. III discusses the proposed method of content-adaptive image steganography, Sect. IV presents a numerical example for embedding and extraction, Sect. V details the experimental setup, while Sect. VI concludes the paper.

II. RELATED WORK

A method to limit the steganographic distortion by utilizing an adjustable data hiding algorithm is presented in [21]. The method starts by dividing the image into 2x2 overlapping blocks. Each block is expanded to a 3x3 block and the empty pixels are interpolated using the four corner pixels. Once the whole cover image is interpolated, this expanded image is considered for data embedding. The cover image is again divided into 2x2 sized and this time non-overlapping blocks. A 2x2 pixel patch consists of a corner pixel and three embeddable pixels. The difference between the embeddable pixels and a corner pixel is calculated and three difference values are obtained. A secret payload size is determined for each pixel by taking the binary logarithm of respective differences.

To limit the distortion to an acceptable level, the secret bits are taken to be no more than a maximum value n. The message size and the parameter n are embedded in the cover image as side information after the secret message is embedded. This side information is required to extract the hidden information at the receiver's side. When stego image is received at the other end, the interpolated values are recalculated. The decimal equivalent of secret bits is extracted by taking the difference between interpolated and embedded pixels. The number of secret bits embedded per pixel is determined using the parameter n and the hidden data is retrieved by calculating the binary equivalent values. The bits are concatenated to form the secret message stream. The steps are repeated until the whole secret message is extracted. One limitation to the above method is that the pixel variation is not analyzed in all possible ways. The pixel difference should have been considered in all three directions, since the interpolated values are known at the receiver's end.

An adaptive steganography method is presented in [22] which is based on block complexity estimation and matrix embedding. An input cover image is considered to be composed of small non-overlapping segments of size 2x4. Each local segment is then converted to a 1D representation to obtain seven overlapping pixel pairs. The pixel difference computation is performed for each pixel pair and difference outcomes are obtained for an individualized segment. A total of eight complexity levels are defined, and an embedding strategy is set for each level based on the payload length. Consequently, only the higher complexity levels are utilized if the payload length is small. A secret payload is estimated for each level by solving an optimization problem. Efficient data hiding method such as matrix embedding is then utilized in the data embedding step. One limitation of the above scheme is that, in a 1D local segment, the difference of non-corner pixels is computed in two directions while the corner pixel's difference is computed in a single direction.

An efficient edge-adaptive embedding algorithm is presented in [23] which is based on XOR coding. The edge detection method allows for preserving the edge intensity map before and after embedding. Initially, an image is partitioned into 3x3 non-overlapping blocks, and the pixel difference is exploited in three directions e.g., horizontal, vertical and diagonal using only the corner pixels. A maximum edge score from the four computed edges is assigned to the block. After the assignment of edge scores, the blocks are then arranged in the order of the edge score from highest to lowest. A threshold is selected adaptively based on the payload length which determines the number of blocks that are used for data embedding. The four pixels (other than corner pixels) are used for data embedding. The four pixels are paired in an adjacent manner such that three pairs are formed. XOR operation is performed between the LSBs of pixels in each pair and three binary results are achieved. The three binary outcomes are compared to three message bits using XOR operation. A mapping table containing eight combinational possibilities of matching is used to switch the LSB of pixels such that it represents the message bits. The XOR embedding scheme is also extended to edge adaptive version where the mean edge intensity of the block determines the number of data bits to be embedded in a single pixel. Noticeably, the scheme utilizes only four pixels in the edge intensity calculation however the central pixel can also be included in the calculation. Therefore, the scheme does not provide an accurate identification of texture complexity. Moreover, only the four pixels are utilized in embedding of data and the fifth pixel is left unembedded therefore wasting the embedding capacity. The use of mean edge intensity is not appropriate way to determine the bits per pixel since two blocks having same average difference may have different complexity levels. The above arguments are validated in the experimental results section.

An adaptive steganography technique based on Tree-Based Parity Check (TBPC) is proposed in [24]. In a cover image, every two adjacent pixels are paired to form 1x2 sized non-overlapping blocks. The absolute pixel difference in an individualized block is then computed. Six kinds of blocks are defined and each block is evaluated to qualify for one out of six complexity levels. Data embedding is then performed on the respective blocks using TBPC embedding algorithm. Noticeably, the use of non-overlapping blocks in the edge-detection step limits the hiding capacity of the scheme.

A content-adaptive image steganography method for color images is presented in [25] which analyses the texture of a 3x3 block of image by means of an energy measuring function based on the Ising spin glass model. The energy function calculates energy of a pixel centered in a 3x3 window in a similar way as applying a point detection kernel on an image segment. The image is masked to represent the 4 higher bit planes before energy calculation so that the data can be accurately recovered in the extraction stage. The authors have achieved embedding capacity of 4bpp and their scheme can withstand the first-order statistical test (dual statistics test and stirmark analysis). However, the design of energy function does not consider the individual pixel correlation with respect to the central pixel. Instead, the energy of a pixel depends on the sum of difference between the eight neighborhood pixels and the central pixel. Thus, the energy outcome may be biased by a high difference value in a single direction.

To summarize, the above techniques are designed for content-adaptive image steganography. However, most of the techniques do not take into account the information of pixel variation around the central pixel in all possible directions and therefore they are unable to accurately analyse the texture complexity. In the light of above discussion, in this paper, an accurate methodology of texture analysis is proposed that provides high embedding capacity, visual quality and security.

III. PROPOSED METHOD

The proposed method starts by dividing the cover image into small overlapping blocks. A complexity identification method is then applied on an individualized pixel block.

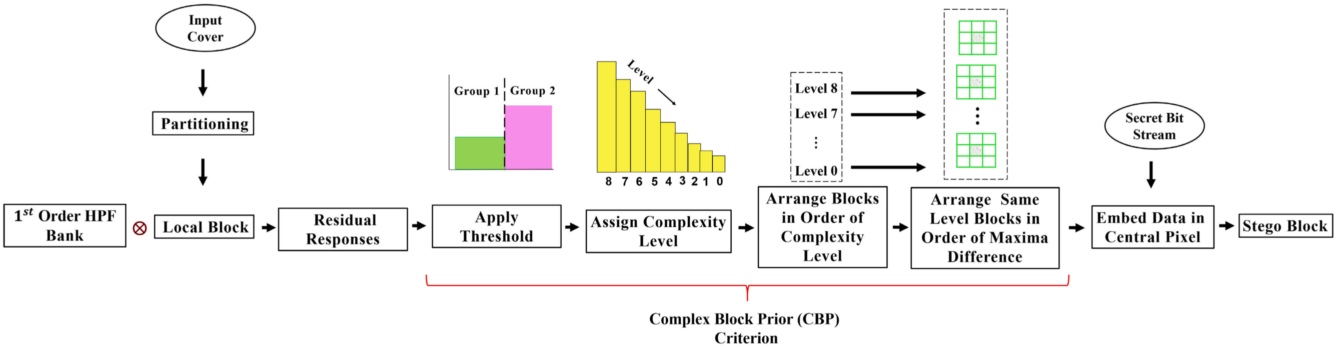


FIGURE 2. The flow diagram of the proposed texture complexity estimation method.

A multi-directional HPF bank is utilized to calculate the pixel variation among the central pixel and all neighboring pixels. A Complex Block Prior (CBP) criterion evaluates the pixel differences and assigns one out of nine complexity levels to a pixel block. The blocks are arranged in descending order of the complexity level and the same level pixel blocks are rearranged in the order of the maximum difference. Data is embedded in the central pixel using a new adaptive algorithm depending on either single or multi-bit embedding. The number of secret bits per pixel is estimated based on the maximum of the pixel differences in a pixel block. The proposed method follows the road map in Fig 2. The details of the presented embedding algorithm are discussed as follows.

A. IMAGE SEGMENTATION

Given an input cover, proceed horizontally to partition the image into small overlapping blocks. Three types of pixel blocks are obtained based on the number of neighboring pixels. The pixels on the periphery of the image are the corner pixels and non-corner pixels. The corner pixels have three neighbors, therefore they form a block of size 2x2 (type 1). The periphery pixels other than the corner pixels have five neighbors, therefore they form a block of size 2x3 (type 2) on the horizontal edge and block of size 3x2 (type 2) on the vertical edge. The non-periphery pixels have eight neighbors, therefore a block of size 3x3 (type 3)is formed. Fig. 3 shows the three types of pixel blocks. The pixel with a blue dot represents the central pixel. Please note that the image is partitioned into overlapping blocks so that each pixel can take part in embedding and the embedding capacity can be improved. Given a pixel block, the next step is to identify the texture complexity or the pixel variation occupied by the central pixel.

B. TEXTURE COMPLEXITY ESTIMATION

To identify the degree of complexity occupied by a central pixel in a local region, the underlying assumption is: In a local region, the central pixel must vary (to a certain degree) in all directions. The desirable variation improves the visual quality of stego image and reduces the probability of the pixel to be estimated by statistical steganalysis attacks [26]. In the light of the above assumption, the proposed method aims to calculate the pixel variation with respect to all neighbors. A 3x3 block of pixels is shown in Fig. 4. The difference of central pixel p_c is calculated with respect to its eight neighbors (p_i, $i = \{1, \cdots 8\}$). The eight central differences (d_{ci}s) are grouped under a difference set D as shown in (1). This set plays a significant role in determining the complexity of a pixel block. For the type 1 and 2 blocks, the central differences are also taken to be eight in number. The differences for the neighbors that do not exist in the two types of blocks are taken as zero. The methodology of the computation of the pixel variation in multiple directions is represented in the proceeding sub-section.

$$D = [d_{c1}, d_{c2}, d_{c3}, \cdots d_{c8}]. \tag{1}$$

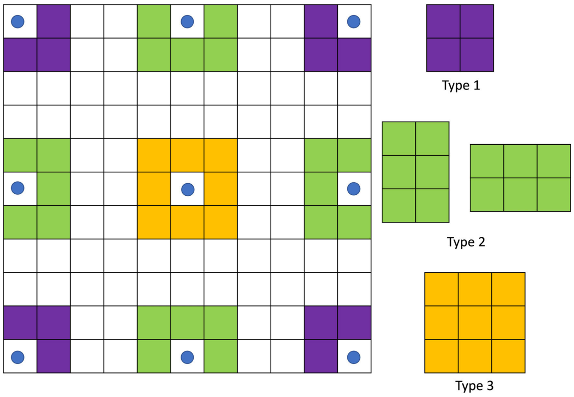


FIGURE 3. An 11x11 image representing the three types of pixel blocks. The pixel blocks are classified based on the number of neighbors of the central pixel. The central pixel is represented by a blue dot.

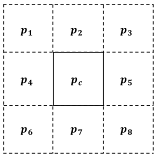
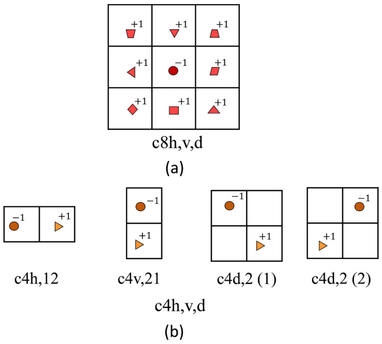


FIGURE 4. A 3x3 pixel block representing central (solid border) and neighboring (dashed borders) pixels.

1) GENERATE RESIDUAL RESPONSES

Before discussing the methodology, it is important to mention that the computation of pixel variation is performed only on the higher bit planes of the cover image. In this way, the texture complexity remains invariant before and after embedding and data can be accurately extracted based on the complexity of the region. The number of higher bit planes for texture complexity computation is calculated based on a parameter ε which defines the maximum number of secret bits embedded per pixel. Since in our case the total number of bit planes is eight, therefore given a value of ε then the number of higher bit planes h for the calculation of residual responses is calculated as:

$$h = 8 - \varepsilon \tag{2}$$



(a)

(b)

FIGURE 5. High pass filter bank (a) the theoretical eight high pass filters required to compute pixel difference in eight directions (b) the original four high pass filters to compute pixel difference in eight directions.

Now, for every pixel block, the central pixel variation is exploited in all possible directions using a high pass filter (HPF) bank. Fig. 5(a) presents the eight 3x3 sized high pass filters which are built as first-order linear filters. The central pixel p_c at which a filter is evaluated is marked with a dot and paired with a symbol representing the neighboring pixel. Since in a 3x3 block, there are eight neighboring pixels therefore, a total of eight filters are formed. The integer (-1) accompanying the dot represents the order of the filter. This is a theoretical representation of the HPFs to compute the pixel difference in eight directions. However, for efficient filtering only four HPFs with smaller size are required to generate the eight residual responses. The sliding property of a filter allows each filter to compute pixel difference along two directions as shown later in (1). These four HPFs are presented under the set c4h,v,d in Fig.5(b). The notation for the HPF bank is FnDs, where F specifies the central pixel e.g., central (c). n represents the number of filters in the filter bank. For example, there are eight filters in the c8h,v,d and four filters in the c4h,v,d filter bank. The notation D specifies the direction in which the pixel difference is calculated. For example, c4h,v,d computes central pixel difference in three directions e.g., horizontal (h), vertical (v) and diagonal (d). The additional notation s in the sub-filters of c4h,v,d denotes the size of filter e.g., in c4h,12 the filter size is 1x2 and in c4d,2 the filter is a square matrix so a single 2 represents the size 2x2. The application of HPF bank on a cover image I is represented mathematically as follows:

$$\begin{aligned} M_1 &= \left| I \circledast H_{c4h,12} \right| for \quad i = \{4, 5\}. \\ M_2 &= \left| I \circledast H_{c4v,21} \right| for \quad i = \{2, 7\}. \\ M_3 &= \left| I \circledast H_{c4d,2(1)} \right| for \quad i = \{1, 8\}. \\ M_4 &= \left| I \circledast H_{c4d,2(2)} \right| for \quad i = \{3, 6\}. \end{aligned} \tag{3}$$

where $\circledast$ denotes the convolution operation and H_x represents a sub-filter x belonging to set c4h,v,d. Equation (3) results in four residual matrices M_ns, each providing two difference values (d_{ci}) between the central pixel and the ith neighboring pixel in the block. Next, we prioritize the blocks for embedding by assigning complexity levels based on a sign function and arranging the blocks in each level by analyzing the residual responses. This is performed by our proposed Complex Block Prior (CBP) criterion as described in next section.

2) COMPLEX BLOCK PRIOR (CBP) CRITERION

By the definition, if the pixels in a block have high local variation then a high complexity level is assigned to the block. On the contrary, if the pixel variation is less, a low complexity level is assigned to the block. The CBP criterion uses a threshold th_1 to divide the eight residual responses into two groups. The number of residual responses in a group then determines the complexity level of the pixel block. Consider a vector D containing the eight residual responses. Then the complexity level γ is calculated as defined in (4).

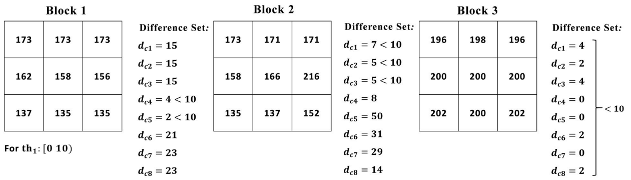
$$\gamma = \sum_{i=1}^{8} f(d_{ci} - th_1). \tag{4}$$

where f(.) is a sign function and is defined as follows.

$$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{5}$$

As an example, suppose D = [15, 10, 12, 4, 2, 21 22, 23] and $th_1 = 10$ then the complexity level $\gamma = 1 + 1 + 1 + 0 + 0 + 1 + 1 + 1 = 6$. Here, the threshold th_1 plays the role of an adjustment scale. By changing th_1 the complexity level of a block is changed. If th_1 increases, more and more pixel blocks jump from higher complexity levels to lower levels. On the other hand, if th_1 decreases, more and more pixels jump from lower complexity levels to higher levels. Therefore, for better accuracy, we keep the value of th_1 moderate e.g., between 8 and 12, not too high and not too low. From (4), if all eight residual responses are greater than or equal to th_1 then a complexity level of 8 is assigned to the pixel block. Similarly, a complexity level of 0 is assigned to the block whose all responses are less than th_1. Therefore, there are a total of nine complexity levels.

Fig. 6 shows three example pixel blocks, each representing a different complexity level. After the complexity level assignment, the blocks are arranged in the order of γ



Block 1

173	173	173
162	158	156
137	135	135

For th$_1$: [0 10)

Difference Set:
$d_{c1} = 15$
$d_{c2} = 15$
$d_{c3} = 15$
$d_{c4} = 4 < 10$
$d_{c5} = 2 < 10$
$d_{c6} = 21$
$d_{c7} = 23$
$d_{c8} = 23$

Block 2

173	171	171
158	166	216
135	137	152

Difference Set:
$d_{c1} = 7 < 10$
$d_{c2} = 5 < 10$
$d_{c3} = 5 < 10$
$d_{c4} = 8$
$d_{c5} = 50$
$d_{c6} = 31$
$d_{c7} = 29$
$d_{c8} = 14$

Block 3

196	198	196
200	200	200
202	200	202

Difference Set:
$d_{c1} = 4$
$d_{c2} = 2$
$d_{c3} = 4$
$d_{c4} = 0$
$d_{c5} = 0$
$d_{c6} = 2$
$d_{c7} = 0$
$d_{c8} = 2$
< 10

FIGURE 6. Three blocks with complexity level 7, 6 and 0, respectively.

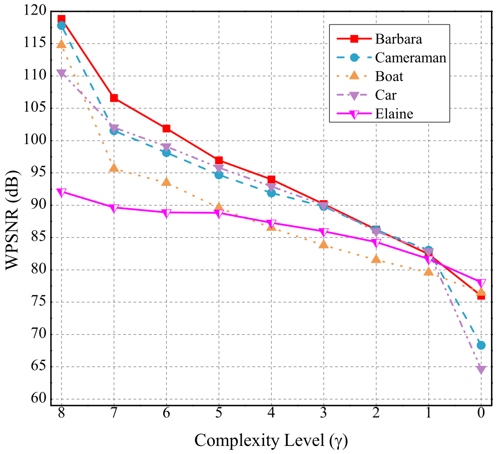


FIGURE 7. The visual quality (measured in terms of WPSNR) analyzed on five standard images e.g., Barbara, Cameraman, Boat, Car, and Elaine, embedding a fixed payload into pixels of each γ separately.

form level 8 to level 0. This is to prioritize the higher-level blocks for embedding since embedding in the highly complex blocks provides good visual quality of stego image along with the high-security performance in feature steganalysis. For better insight, consider embedding in the pixels of a specific complexity level (γ). Fig. 7 shows the visual quality (measured in terms of WPSNR) analyzed on five standard images e.g., Barbara, Cameraman, Boat, Car and Elaine when embedding a fixed payload into pixels of each γ separately. Noticeably, the visual quality reduces as the γ of the pixel blocks decreases. Therefore, we prefer the higher-level blocks for embedding.

Another step which needs to be taken for the better visual quality of the stego image is to rearrange the pixel blocks belonging to the same complexity level in descending order of the maximum residual response. This is to prioritize the same-level pixel blocks with the highest pixel variation for embedding. The maximum residual response D_{max} of a pixel block is calculated as follows.

$$D_{max} = max(d_{c1}, d_{c2}, d_{c3}, \cdots d_{c8}) \tag{6}$$

where max (.) is a maximum operator. Here it is important to mention that the sorting of same-level blocks is not required for all the levels. The selection of a complex level for such sorting depends on the size of secret message. If all pixels associated with a complex level are not utilized for embedding, then we sort the pixel blocks to prefer the highest variation blocks for embedding. Hence, the method is efficient in application as well. Table 1 presents the comparison of the visual quality of stego images produced with sorted blocks vs unsorted blocks. Noticeably, the visual quality results using the sorted blocks are better than the ones produced using the unsorted blocks.

TABLE 1. Comparison of visual quality of stego images produced with sorted blocks vs unsorted blocks.

Images	WPSNR (dB)		
	Sorted Pixels	Unsorted Pixels	Difference
Barbara	61.70	60	1.7
Cameraman	65.92	99.10	1.92
Boat	66.78	65.7	1.08
Car	65.51	64.19	1.32
Elaine	65.44	63.79	1.65
Average	65.05	63.53	1.51

Next step is the data embedding step.

3) DATA EMBEDDING

This section presents a new adaptive data embedding algorithm which is designed to select a hiding scheme based on either single or multibit embedding and a capacity limiting parameter ε. The flow diagram of the adaptive algorithm is presented in Fig. 8. The input to this algorithm is a set of sorted pixel blocks generated form the proposed CBP algorithm along with the maximum residual response D_{max} of each block. The embedding procedure starts as follows:

1) The blocks are processed in sequence from top to bottom and we estimate the number of bits to be embedded per pixel. This is done by taking log-base 2 of the D_{max}. The bits per pixel a estimation is expressed mathematically as follows.

$$a = \lfloor \log_2 D_{max} \rfloor \tag{7}$$

where ⌊ x ⌋ outputs the biggest integer no larger than x e.g., floor. It is important to mention that the mean of the residuals as used previously in [5] is not appropriate for estimation of bits per pixel since two blocks having same average difference intensity may carry different pixel complexity. It is worthwhile mentioning here that this observation has not been discussed before in related techniques.

2) The parameter ε is introduced in the equation which acts as a capacity limiter and is used to set an upper bound on a. Thus Eq. 2 is modified as:

$$a = min(\lfloor \log_2 D_{max} \rfloor, \varepsilon) \tag{8}$$

where min(.) is a minimum operator. The parameter ε can take any value between 1 and 8. Since we have used an 8-bit image representation therefore, a maximum of eight bits and minimum of one bit can be embedded in a single pixel. The visual quality of stego image is maximum at $\varepsilon = 1$ and it reduces as the ε is increased.

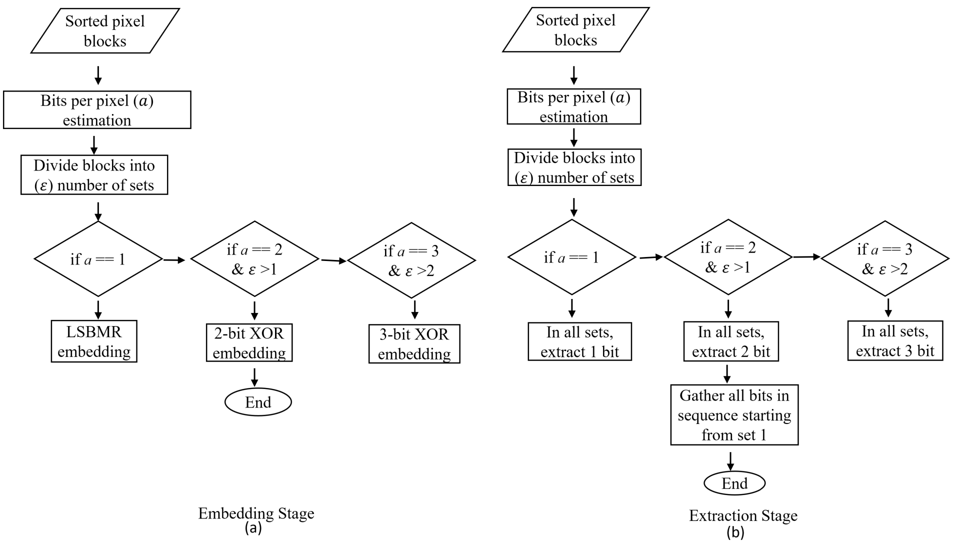

Sorted pixel blocks
Bits per pixel (a) estimation
Divide blocks into (ε) number of sets
if $a == 1$
if $a == 2$ & $\varepsilon > 1$
if $a == 3$ & $\varepsilon > 2$
LSBMR embedding
2-bit XOR embedding
3-bit XOR embedding
End
Embedding Stage
(a)
Sorted pixel blocks
Bits per pixel (a) estimation
Divide blocks into (ε) number of sets
if $a == 1$
if $a == 2$ & $\varepsilon > 1$
if $a == 3$ & $\varepsilon > 2$
In all sets, extract 1 bit
In all sets, extract 2 bit
In all sets, extract 3 bit
Gather all bits in sequence starting from set 1
End
Extraction Stage
(b)

FIGURE 8. The flow diagram of the embedding and extraction procedure.

One can increase the visual quality at the expense of smaller embedding capacity.

3) Once the number of secret bits per pixel a is estimated for all the blocks, we now divide the central pixels into ε number of sets. For example, if $\varepsilon = 3$ then the pixels are divided into three sets. Each set contains the pixels with the same a.

4) Given a set, gather the secret bits from the secret bit stream. Replace the least significant bits LSBs of the central pixel using highly efficient embedding scheme. Since the visual quality of a stego image also depends on the modification rate of the embedding algorithm, therefore we present two efficient embedding schemes which provide a low modification rate for single and multibit embedding. Although the use of efficient embedding further improves the visual quality of the stego image. However, in section 4, we demonstrate the effectiveness of our proposed texture complexity estimation method by using simple LSB replacement embedding which does not provide a reduced modification rate. The adaptive embedding proceeds as follows.

For the set with $a = 1$, the embedding is performed in the using the subtractive relation of a pixel pair. For this, arrange the pixels of set 1 and onwards in adjacent pairs. In each pair, the first secret bit is embedded in the first pixel and the second secret bit is embedded in the subtractive relation of the two pixels. Suppose m_1 and m_2 are the two secret bits and q_1 and q_2 are the LSBs of the two cover pixels x_1 and x_2. Then the subtractive relation X of the two pixels is taken as follows:

$$X = \begin{cases} \left\lfloor \left| \frac{x_1}{2} - x_2 \right| \right\rfloor & if x_1 \geq 2x_2 \\ \left\lceil \left| \frac{x_1}{2} - x_2 \right| \right\rceil & if x_1 < 2x_2 \end{cases} \tag{9}$$

The LSB of the subtractive relation X is $\mathrm{q} = \mathrm{LSB}[\mathrm{X}]$. The embedding is followed as:

Case 1:
if $\mathrm{q}_1 = \mathrm{m}_1$ and $\mathrm{q} = \mathrm{m}_2$ no change in cover pixels required.

Case 2:
if $\mathrm{q}_1 = \mathrm{m}_1$ and $\mathrm{q} \neq \mathrm{m}_2$, change x_2 as $\mathrm{x}_2' = \mathrm{x}_2 \pm 1$

Case 3: if $q_1 \neq m_1$ and $q = m_2$ change x_1 as

$$x_1' = \begin{cases} x_1 + 1 & \text{if} \quad x_1 \quad \text{is} \quad \text{even} \quad \text{or} \quad \text{zero} \\ x_1 - 1 & \text{if} \quad x_1 \quad \text{is} \quad \text{odd} \end{cases}$$

Case 4: if $q_1 \neq m_1$ and $q = m_2$, change x_1 as

$$x_1' = \begin{cases} x_1 + 1 & \text{if} \quad x_1 \quad \text{is odd} \\ x_1 + 2 & \text{if} \quad x_1 \quad \text{is zero} \\ x_1 - 1 & \text{if} \quad x_1 \quad \text{is even} \end{cases}$$

Embedding in this manner reduces the average modification form 5/4 bits of LSBR to 3/4 bits. Therefore, further improvement in visual quality is achieved. The embedding continues until the end of secret bits or until all the pixels in set 1 have been utilized.

For the sets with $a > 1$, the multibit embedding scheme using the XOR coding is utilized for data hiding. First, the condition of ε is checked. If $\varepsilon = 2$ then the pixels of set 2 and onwards are utilized for 2-bit embedding. If $\varepsilon = 3$ then the pixels of set 2 are utilized for 2-bit embedding and the pixels of set 3 and onwards are utilized for 3-bit embedding and so on. Multibit XOR embedding uses four cover pixels to embed three secret bits in each of the bit planes. We arrange the pixels in set 2 and onwards in a group of four. The four cover pixels in each group are arranged in three pairs and XOR operation is performed between the bits of each of three-pixel pairs using the following equations.

$$\begin{aligned} k_{1p} &= q_{1p} \circledast q_{2p}. \\ k_{2p} &= q_{3p} \circledast q_{4p}. \\ k_{3p} &= q_{1p} \circledast q_{3p}. \end{aligned} \tag{10}$$

where, p represents the bit plane from which the bits are considered. The resulting bits k_{1p}, k_{2p} and k_{1p} for the pth bit plane are compared with the three secret bits m_1, m_2 and m_3. A mapping table is used to decide the cover bit modification. The mapping table is shown in Fig. 9. XOR embedding is applied to each group until the end of secret bits or until all the groups have been utilized for embedding. The average modification is reduced form 1.5 bits of LSBR to 1.25 bits.

Case			Cover bits alteration
$m_1 = k_1$	$m_2 = k_2$	$m_3 = k_3$	No change
$m_1 = k_1$	$m_2 = k_2$	$m_3 \neq k_3$	Complement q_3 and q_4
$m_1 = k_1$	$m_2 \neq k_2$	$m_3 = k_3$	Complement q_4
$m_1 = k_1$	$m_2 \neq k_2$	$m_3 \neq k_3$	Complement q_3
$m_1 \neq k_1$	$m_2 = k_2$	$m_3 = k_3$	Complement q_2
$m_1 \neq k_1$	$m_2 = k_2$	$m_3 \neq k_3$	Complement q_1
$m_1 \neq k_1$	$m_2 \neq k_2$	$m_3 = k_3$	Complement q_2 and q_4
$m_1 \neq k_1$	$m_2 \neq k_2$	$m_3 \neq k_3$	Complement q_1 and q_4

FIGURE 9. The mapping table of XOR embedding defining the embedding strategy.

In order to enable the receiver to correctly extract the secret bits, the size of secret bit stream M, the limiting parameter ε and the threshold th$_1$ is sent alongside the stego image. This side information synchronization can be achieved by embedding four bytes concatenated at the start of the secret message using four pixels.

4) EXTRACTION STAGE

The extraction of secret data is simpler and more efficient than the embedding process. The proposed algorithm allows the extraction of data without the need for an original cover image. Fig. 7 represents the flow diagram of the data extraction process. Given the stego image I_s which results from the embedding algorithm described earlier, the algorithm starts by dividing the image into overlapping blocks in the same fashion and manner as described in the embedding step. The message length, the value of parameter ε and the threshold th$_1$ is retrieved from the four starting pixel blocks. The number of higher bit planes of the stego image are then calculated using (2) as $h = 8 - \varepsilon$. After masking the lower bit planes of the stego image, the texture complexity of each block is estimated using the proposed texture complexity estimation method. The HPF bank c4h,v,d as shown in fig 3(b) is applied on the masked image and the residual matrices are generated as follows.

$$\begin{aligned} M_1 &= \left|I \circledast H_{c4h,12}\right| for \quad i = \{4, 5\}. \\ M_2 &= \left|I \circledast H_{c4v,21}\right| for \quad i = \{2, 7\}. \\ M_3 &= \left|I \circledast H_{c4d,2(1)}\right| for \quad i = \{1, 8\}. \\ M_4 &= \left|I \circledast H_{c4d,2(2)}\right| for \quad i = \{3, 6\}. \end{aligned} \tag{11}$$

The four residual matrices M$_{cn}$s, each providing two difference values (d$_{ci}$) between the central pixel and the ith neighboring pixel in the block. A complexity level out of nine levels is assigned to each of the pixel blocks using (4).

The CBP algorithm assigns the same complexity level to each pixel block as was assigned previously in the embedding step. Therefore, the texture complexity of all the pixel blocks remains exactly invariant after data embedding. The CBP algorithm arranges the pixel blocks in the order of complexity level from highest to lowest and the same level pixels are sorted according to their maximum residual difference D$_{max}$. We know the secret message size therefore, only the pixels of the complexity level for which the secret message ends amid will be considered for sorting. Now, the data extraction proceeds as follows:

1) The blocks from the complexity identification step are processed in a sequence starting from top to bottom and the number of secret bits per pixel for each block is estimated based on its D$_{max}$.
2) The central pixels are divided into ε number of sets where ε is obtained from the previous parameter recovery stage. Each set contains pixels with the same number of secret bits per pixel a. For the set with $a = 1$, arrange the pixels of set 1 and onwards in adjacent pairs. In each pair, the first secret bit is retrieved from the first pixel and the second secret bit is retrieved from the subtractive relation of the two pixels. The extraction continues until all the secret bits in set 1 have been extracted.

   For the sets with $a > 1$, the condition of ε is checked. If $\varepsilon = 2$ then the 2-bits are extracted from the pixels of set 2 and onwards. If $\varepsilon = 3$ then the 2-bits are extracted from the pixels of set 2 and 3-bits are extracted from the pixels of set 3 and onwards and so on. We arrange the pixels in set 2 and onwards in a group of four. The four cover pixels in each group are arranged in three pairs. The three secret bits are retrieved by performing XOR operation between the bits of pth bit plane of each of three-pixel pairs as followed by the following equations.

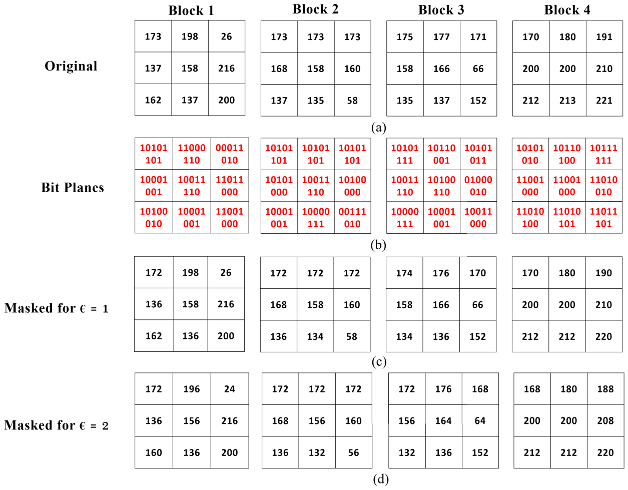
$$\begin{aligned} m_1 &= q_{1p} \circledast q_{2p}. \\ m_2 &= q_{3p} \circledast q_{4p}. \\ m_3 &= q_{1p} \circledast q_{3p}. \end{aligned} \tag{12}$$

   XOR embedding is applied to each group until the end of secret bits.
3) Concatenate the secret bits extracted from the sets in the following sequence.
   (bits from set 1) ‖ (bits from set 2) ‖ (bits form set 3) $\cdots$

IV. NUMERICAL EXAMPLE OF EMBEDDING AND EXTRACTION

Consider an example for the illustration of the proposed content-adaptive embedding scheme. Suppose a cover image is partitioned into overlapping blocks as shown in Fig. 10(a).

For convenience only type 1 pixel blocks are considered. The pixel intensities of the four blocks are presented in binary form in Fig 10(b). As discussed earlier, the residual responses are generated by using only the (8 - ε) higher bit planes. Here, two embedding cases are presented e.g., for $\varepsilon = 1$ and $\varepsilon = 2$. The pixel blocks for the two cases are presented in Fig 10(c) and (d), respectively. Please note that the embedding scenario is presented for the same complexity-level γ when $th_1 = 8$ and is verified in the proceeding section.



	Block 1			Block 2			Block 3			Block 4		
Original	173	198	26	173	173	173	175	177	171	170	180	191
	137	158	216	168	158	160	158	166	66	200	200	210
	162	137	200	137	135	58	135	137	152	212	213	221

(a)

Bit Planes	10101 101	11000 110	00011 010	10101 101	10101 101	10101 101	10101 111	10110 001	10101 011	10101 010	10110 100	10111 111
	10001 001	10011 110	11011 000	10101 000	10011 110	10100 000	10011 110	10100 110	01000 010	11001 000	11001 000	11010 010
	10100 010	10001 001	11001 000	10001 001	10000 111	00111 010	10000 111	10001 001	10011 000	11010 100	11010 101	11011 101

(b)

Masked for ϵ = 1	172	198	26	172	172	172	174	176	170	170	180	190
	136	158	216	168	158	160	158	166	66	200	200	210
	162	136	200	136	134	58	134	136	152	212	212	220

(c)

Masked for ϵ = 2	172	196	24	172	172	172	172	176	168	168	180	188
	136	156	216	168	156	160	156	164	64	200	200	208
	160	136	200	136	132	56	132	136	152	212	212	220

(d)

FIGURE 10. Example of four pixel blocks (a) original pixel blocks, (b) Bit planes (c) Pixel intensities for $\varepsilon = 1$ and (d) Pixel intensities for $\varepsilon = 2$.

A. SINGLE-BIT EMBEDDING WITH $\varepsilon = 1$

When $\varepsilon = 1$, the first seven higher bit planes are used for the calculation of residual responses. The residual resposes for the four cover blocks are calculated using (3). The difference vector D_n for n^{th} block is thus D_1 X $\{14, 40, 132, 22, 58, 4, 22, 42\}$, $D_2 = \{14, 14, 14, 10, 2, 22, 24, 100\}$, $D_3 = \{8, 10, 4, 8, 100, 32, 30, 14\}$ and $D_4 = \{30, 20, 10, 0, 10, 12, 12, 20\}$, respectively. The complexity level γ_n when $th_1 = 8$ for n^{th} block can then be computed using (4). Hence $\gamma_1 = 1+1+1+1+1+0+1+1 = 7$, $\gamma_2 = 1+1+1+1+0+1+1+1 = 7$, $\gamma_3 = 1+1+0+1+1+1+1+1 = 7$, and $\gamma_4 = 1+1+1+0+1+1+1+1 = 7$, for n $=\{1, 2, 3, 4\}$. Now, we obtain the maximum residual difference for n^{th} block as: $D_{max,1} = 132$, $D_{max,2} = 100$, $D_{max,3} = 100$ and $D_{max,4} = 30$. The pixel blocks are arranged in the order of maximum residual difference D_{max}. Next step is to calculate the bits per pixel a using (8) as follows:

$$a_1 = min(\lfloor \log_2 132 \rfloor, 1) = 1$$
$$a_2 = min(\lfloor \log_2 100 \rfloor, 1) = 1$$
$$a_3 = min(\lfloor \log_2 100 \rfloor, 1) = 1$$

and

$$a_4 = min(\lfloor \log_2 30 \rfloor, 1) = 1$$

Following the flow diagram of Fig. 8(a), since all blocks have same a, the pixel blocks are kept in set 1 and arranged in adjacent pairs. Now the embedding procedure starts as follows: Given a message sequence M $= \{1\ 0\ 0\ 0\ 1\ 0\}$. Now, consider the first pair of central pixels $(x_1, x_2) = (158, 158)$. Thus, $q_1 = 0$. The subtractive relation X is calculated as follows:

$$X = \left\lceil \left| \frac{158}{2} - 158 \right| \right\rceil = 77$$

Here q $=$ LSB[77] $= 1$. Comparing the cover bits with message bits it is concluded that $q_1 \neq m_1 \neq 1$ and $q \neq m_2 \neq 0$. Therefore, case 4 is followed and $x'_1 = x_1 - 1 = 158 - 1 = 157$. Therefore, embedding of two bits require modification of single cover pixel. Similarly embedding is performed for next pixel pair as follows. Given the second pixel pair $(x_1, x_2) = (166, 200)$. Thus, $q_1 = 0$. The subtractive relation X is calculated as follows:

$$X = \left\lceil \left| \frac{166}{2} - 200 \right| \right\rceil = 117$$

Here q $=$ LSB[117] $= 1$. Comparing the cover bits with message bits it is concluded that $q_1 = m_1 = 0$ and $q = m_2 \neq 1$. Therefore, case 2 is followed and $x'_2 = x_2 - 1 = 200 - 1 = 199$. At the data extraction stage, since only the higher bit planes are used for residual responses therefore, the complexity level remains invariant and the blocks are arranged in similar manner as at the trasmitters end. The blocks for set 1 are arranged in pairs and the message bits are recovered from the subtractive relation of pixels.

B. MULTI-BIT EMBEDDING WITH $\varepsilon = 2$

In the case with $\varepsilon = 2$, the maximum allowable embedding bits per pixel is 2. Therefore, 2-bit embedding is performed. The residual responses are calculated using the first 8 - 2 = 6 bit planes. The pixel blocks for residual responses are shown in Fig. 10(d). The difference vector D_n for n^{th} block is thus $D_1 = \{16, 40, 132, 20, 60, 4, 20, 44\}$, $D_2 = \{16, 16, 16, 12, 4, 20, 24, 100\}$, $D_3 = \{8, 12, 4, 8, 100, 32, 28, 12\}$ and $D_4 =\{32, 20, 12, 0, 8, 12, 12, 20\}$, respectively. The complexity level γ_n and $th_1 = 8$ is same as in first case. Similarly, the D_{max} is also same as in case 1 and thus the arrangement of pixels is also same. Next step is to calculate the bits per pixel a using (8) as follows:

$$a_1 = min(\lfloor \log_2 132 \rfloor, 2) = 2$$
$$a_2 = min(\lfloor \log_2 100 \rfloor, 2) = 2$$
$$a_3 = min(\lfloor \log_2 100 \rfloor, 2) = 2$$

and

$$a_4 = min(\lfloor \log_2 30 \rfloor, 2) = 2$$

Following the flow diagram of Fig. 8(a), since all blocks have same a, the pixel blocks are kept in set 2 and arranged in a group. Given the message sequence M $= \{1\ 0\ 0\ 0\ 1\ 0\}$, the embedding procedure starts as follows: The capacity for each central pixel is 2 bits therefore, 2-bit XOR embedding is used. The procedure of embedding presented in sub-section 4 is followed as follows: Given the four cover pixels

TABLE 2. Embedding capacity comparison of the proposed method with the technique [21].

Images	Embedding Capacity (EC)					
	n = 3		n = 2		n = 1	
	Proposed	[21]	Proposed	[21]	Proposed	[21]
Lena	582,920	207,567	384,024	174,205	258,988	110,000
Peppers	363,670	204,636	277,098	174,110	205,842	113,027
Baboon	609,980	400,798	379,092	306,337	256,282	168,666
Couple	610,120	277,856	383,214	228,960	257,794	137,598
Boat	614,210	273,615	383,862	222,253	258,222	132,309
Plane	484,480	201,671	356,514	163,837	254,484	101,943
Barbara	577,980	266,801	378,276	217,799	256,488	131,985
Average	549,051	261,849	363,154	212,500	249,729	127,933

$x_1 = 156$, $x_2 = 156$, $x_3 = 164$ and $x_4 = 200$. The two bits from the lower bit planes thus are: $q_{11} = 0$, $q_{12} = 1$, $q_{21} = 0$, $q_{22} = 1$, $q_{31} = 0$, $q_{32} = 1$, $q_{41} = 0$ and $q_{42} = 0$. Now, perform XOR operation among the bits of cover pixels in the first bit plane ($p = 1$).

$$\begin{aligned} k_{11} &= q_{11} \circledast q_{21} = 0 \circledast 0 = 0 \\ k_{21} &= q_{31} \circledast q_{41} = 0 \circledast 0 = 0 \\ k_{31} &= q_{11} \circledast q_{31} = 0 \circledast 0 = 0 \end{aligned}$$

Similarly, the XOR operation among bits of second bit plane ($p = 2$) is performed as:

$$\begin{aligned} k_{12} &= q_{12} \circledast q_{22} = 1 \circledast 1 = 0 \\ k_{22} &= q_{32} \circledast q_{42} = 1 \circledast 1 = 0 \\ k_{32} &= q_{12} \circledast q_{32} = 1 \circledast 1 = 0 \end{aligned}$$

The resulting six bits are compared with the message bits and the mapping table of Fig. 9 is used to decide for cover bit modification. From the comparison for $p = 1$, it is concluded that $k_{11} = m_1 = 0$, $k_{21} = m_2 = 0$ and $k_{31} = m_3 = 0$. Which leads us to compliment q_{21}. Therefore, $q'_{21} = 1$. Similarly, for $p = 2$ $k_{12} = m_4 = 0$, $k_{22} \neq m_5 \neq 1$ and $k_{32} = m_6 = 0$. This leads us to compliment q_{42} thus $q'_{42} = 0$. At the data extraction stage, since only the higher bit planes have used for residual responses therefore, the complexity level remains invariant and the blocks are arranged in similar manner as at the transmitter's end. The blocks for set 2 are arranged in group of four and the message bits are recovered from the XOR relation of pixels as follows.

$$\begin{aligned} m_1 &= q_{11} \circledast q'_{21} = 0 \circledast 1 = 1 \\ m_2 &= q_{31} \circledast q_{41} = 0 \circledast 0 = 0 \\ m_3 &= q_{11} \circledast q_{31} = 0 \circledast 0 = 0 \\ m_4 &= q_{12} \circledast q_{22} = 1 \circledast 1 = 0 \\ m_5 &= q_{32} \circledast q'_{42} = 1 \circledast 0 = 1 \\ m_6 &= q_{12} \circledast q_{32} = 1 \circledast 1 = 0 \end{aligned}$$

Thus, the extracted message sequence is $M = \{1\ 0\ 0\ 0\ 1\ 0\}$

V. EXPERIMENTAL SETUP AND RESULTS

This section provides a description of the effectiveness of the proposed method in terms of providing high embedding capacity, high image quality and the ability of the proposed method to resist the statistical attacks. In the experimental setup, we take three image datasets. The first image dataset contains a set of standard images from the SIPI dataset and other online sources [27]. These eight standard grayscale images include the Lena, Cameraman and Barbara image. The size of each image is 512x512. The other two datasets are the BOWS2 [28] and BOSSbase dataset [29]. Both datasets contain 10,000 grayscale natural images of size 512x512. The example images from the three datasets are shown in figure 10. The proposed algorithm is implemented in Matlab. A pseudo-random number generator is used to generate a bitstream of secret message. A comprehensive comparison with the latest related techniques [21] is given. In addition we also present the description of effectiveness of the proposed complexity estimation method. The security analysis of the proposed method with related techniques is also presented.

A. EMBEDDING CAPACITY AND IMAGE QUALITY ASSESSMENT

Embedding capacity (EC) is the total number of secret bits embedded in the cover image. EC is an essential unit of measurement since it gives an idea of how the proposed method helps to hide more and more data in the cover image. Table 2 presents the embedding capacity achieved by the proposed method in comparison with [21]. The EC is estimated for seven standard images over varying-parameter n. The proposed method provides highest EC for all values of n. In terms of the percentage increase, the EC of the proposed algorithm is on average 46.46%, 28.73%, and 36.51% higher than [21] for n = 1,2 and 3 respectively. There are two reasons behind such an improved payload capacity: 1) the use of separate bit planes for complexity computation and embedding allows to preserve the pixel differences before and after embedding thereby allowing all the pixels to take part in embedding. 2) The use of variable pixel block size for the periphery pixels further adds 1024 pixels to be used in the embedding process. On the contrary the technique of [21] uses only three out of four pixels for embedding in every 2x2 block. Consequently only 75% of the image is available for embedding and remaining 15% is used for calculation of the number of bits per pixel.

The image quality (IQ) parameters such as PSNR, Weighted PSNR (WPSNR) and SSIM are used to quantify the modification in the stego image with reference to the cover image. The mathematical expression of PSNR is expressed as follows:

$$PSNR = 10\log_{10}\frac{255^2}{E}. \tag{13}$$

where E is the mean square error and is defined as follows.

$$E = \frac{\sum_{h=1}^{H}\sum_{w=1}^{W}(C(h,w)-S(h,w))^2}{H\times W}. \tag{14}$$

The PSNR is measured in dB. For a perfectly distortion less image PSNR is infinity. The weighted PSNR is a modified version of simple PSNR. The WPSNR takes into account the texture content of the image as perceived by the human visual system (HVS) and thus measures the visual quality of the image by considering the relationship of the pixels among each other in a local region. There are many models presented in literature that are designed to measure perceptibility of an image as perceived by HVS. In the presented work, we pursue one of the existing models known as "noise visibility function (NVF)" [16]. The WPSNR is presented in the following equation.

$$WPSNR = 10\log_{10}\frac{255^2\times H\times W}{\sum_{h=1}^{H}\sum_{w=1}^{W}(C(h,w)-S(h,w))^2\times n^2}. \tag{15}$$

where HxW represents the size of the image and NVF n is calculated on the cover image as follows.

$$n(C(h,w)) = \frac{1}{1-\sigma_{h,w}^2} \tag{16}$$

It is clear from the expression of NVF that for a local region centered around pixel (h,w), a high texture content will yield an NVF close to 0 while the smooth texture content will represent an NVF of approximately 1.

Another IQ metric is the Structural SIMilarity index SSIM [30]. It is a measure of the perceptual structure of a stego image with reference to cover image. Low order moments such as mean, variance and correlation coefficient are used to compute structure similarity between an original and distorted image. The underlying assumption is that HVS perceives an image by extracting its structural information. Structure information includes luminance, contrast, and correlation. Luminance is estimated by quantifying the mean of an image. Contrast is estimated by the standard deviation and correlation among the two signals is quantified by using the covariance or the correlation coefficient expression. The following notations are used for the above low order moments e.g. mean as m, standard deviation as std and correlation as corr and are mathematically represented as follows:

The luminance similarity among the cover and stego image can then be represented as:

$$lum(C,S) = \frac{2m_cm_s+c_1}{m_c^2+m_s^2+c_1}. \tag{17}$$

The contrast is represented as:

$$std = \left(\frac{\sum_{i=1}^{N}(C_i-m_c)^2}{N-1}\right)^{1/2}. \tag{18}$$

The contrast similarity expression is then represented as follows:

$$cont(C,S) = \frac{2std_cstd_s+c_2}{std_c^2+std_s^2+c_2}. \tag{19}$$

Similarly, the variation similarity is expressed by the following relation:

$$vari(C,S) = \frac{std_{cs}+c_3}{std_c+std_s+c_3}. \tag{20}$$

where, std$_{CS}$ is given as:

$$std_{cs} = \frac{1}{N-1}\sum_{i=1}^{N}(C_i-m_c)(S_i-m_s). \tag{21}$$

The constants c$_1$, c$_2$, and c$_3$ are equal to (k$_1$xL)^2, (k$_2$xL)^2, and (k$_3$xL)^2, where L is the dynamic range of pixels and k$_1$, k$_2$, and k$_3$ $\ll$ 1. SSIM index is now calculated as:

$$SSIM(C,S) = lum\times cont\times vari. \tag{22}$$

SSIM is calculated in a smaller window of size 8x8 for each pixel in the image. Once the SSIM map is obtained, a mean value of the map is calculated. SSIM ranges between [−1 1], a value of 1 represents the ideal similarity while a value of 0 represents no similarity.

Table 3, 4 and 5 respectively represent the PSNR, WPSNR and SSIM values of the proposed algorithm in comparison with the technique [21]. The performance of the proposed method is evaluated using the EC for [21] given in Table 2. Referring to Table 3, it can be seen that the proposed algorithm achieves a higher value of PSNR in comparison with [21] for all n. An average increase of 5.25 dB, 5.54 dB and 6.05 dB is achieved over [21] for n = 1, 2 and 3 respectively. For n = 1 the proposed embedding algorithm embeds in the cover pixels with a reduced modification rate and therefore further limits the embedding distortion. This setting allows generating a high-quality image as reflected by the highest average PSNR value over [21]. Similarly, for n>1 the proposed embedding algorithm provides the highest average PSNR values over [21].

Table 4 presents the WPSNR metric comparison with the technique [21] over varying n. From the table, it can be seen that the proposed algorithm achieves a higher value of WPSNR in comparison with [21] for all n. An average increase of 5 dB, 8 dB and 5 dB is achieved over [21] for n = 1, 2 and 3 respectively. The high-performance margin of

TABLE 3. PSNR comparison of the proposed method with the technique [21].

Images	PSNR (dB)					
	n = 3		n = 2		n = 1	
	Proposed	[21]	Proposed	[21]	Proposed	[21]
Lena	45	39.37	49.72	44.36	57.26	51.90
Peppers	45.31	39.70	49.74	44.52	57.11	50.19
Baboon	42.10	35.28	47.25	41.26	55.39	49.98
Couple	43.73	37.80	48.50	42.92	56.28	50.04
Boat	43.81	37.68	48.55	43.01	56.45	51.45
Plane	45.21	39.07	50.05	44.55	57.56	51.10
Barbara	43.94	37.87	48.73	43.18	53.56	52.23
Average	44.15	38.11	48.93	43.40	56.23	50.98
Difference	6.05		5.54		5.25	

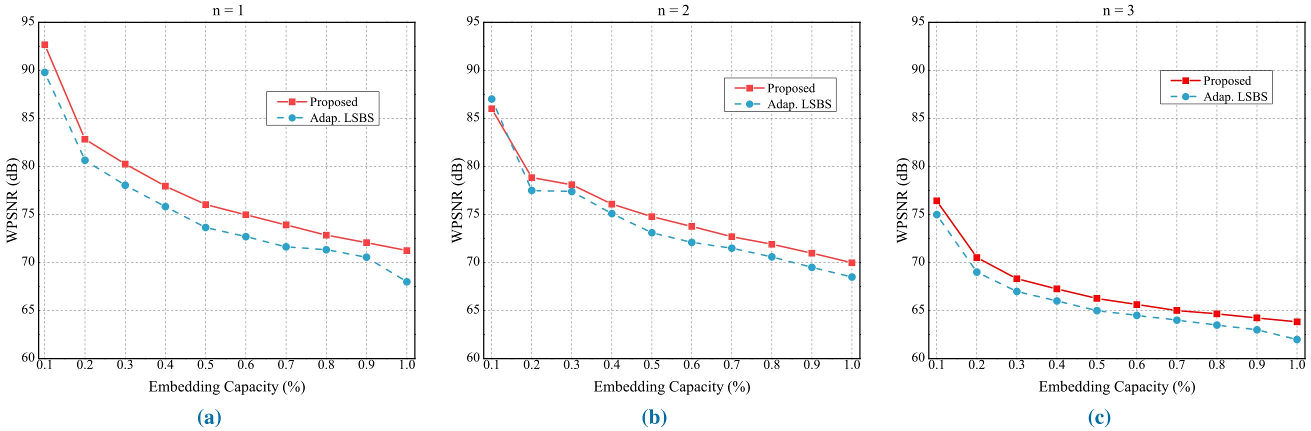
TABLE 4. WPSNR comparison of the proposed method with the technique [21].

Images	WPSNR (dB)					
	n = 3		n = 2		n = 1	
	Proposed	[21]	Proposed	[21]	Proposed	[21]
Lena	61.70	61.79	69.51	64.61	73.76	71.28
Peppers	65.92	60.45	69.29	64.74	71.38	70.72
Baboon	66.78	54.53	75.23	59.75	81.93	67.75
Couple	65.51	59.53	74.93	64.87	80.86	71.05
Boat	65.44	57.63	75.68	62.07	79.21	69.21
Plane	58.52	58.83	63.12	65.05	67.55	74.22
Barbara	63.28	56.40	71.77	61.08	76.38	68.56
Average	63.88	58.45	71.36	63.16	75.87	70.39
Difference	5		8		5	

TABLE 5. SSIM comparison of the proposed method with the technique [21].

Images	SSIM					
	n = 3		n = 2		n = 1	
	Proposed	[21]	Proposed	[21]	Proposed	[21]
Lena	0.9987	0.9943	0.9996	0.9976	0.9999	0.9953
Peppers	0.9998	0.9998	0.9998	0.9975	0.9999	0.9993
Baboon	0.9987	0.9995	0.9995	0.9981	0.9999	0.9998
Couple	0.9993	0.9910	0.9996	0.9967	0.9999	0.9994
Boat	0.9993	0.9917	0.9996	0.9973	0.9999	0.9995
Plane	0.9995	0.9958	0.9996	0.9982	0.9999	0.9994
Barbara	0.9990	0.9847	0.9997	0.9975	0.9999	0.9995
Average	0.9992	0.9921	0.9996	0.9975	0.9999	0.9994
Difference	0.0071		0.0021		0.0005	

WPSNR is achieved since the proposed embedding algorithm adaptively embeds in the highest complex pixels. Moreover the proposed method accurately estimates the embedding capacity per pixel by calculating pixel difference in eight directions and then using the highest difference value for each complexity level while the technique of [21] estimates the embedding capacity by calculating pixel difference in only a single direction. The highest average WPSNR value over [21] displays the effectiveness of the proposed complexity estimation method as well as the reduced modification embedding provided by the embedding step. In order to present the description of the performance of the proposed complexity



(a)

(b)

(c)

FIGURE 11. **The WPSNR performance analysis for the proposed and adaptive LSBS method over (a) n = 1, (b) n = 2 and (c) n = 3.**

estimation step alone, we present the WPSNR comparison of the adaptive n-LSBS with the proposed method. The adaptive n-LSBS uses the proposed complexity identification method to compute the pixel complexity. The only difference lies in the embedding method in the data embedding step which is the simple LSB substitution method. Fig. 11 displays the comparison results over a variety of embedding capacity for $n = [1, 2, 3]$. The results are averaged for the same seven standard test images as mentioned in Table 2. It is clear from the figure that the WPSNR value is reduced with a difference of less than 2.5 dB for $n = 1$ as compared to the proposed algorithm. A difference of less than 1 dB is noticed for $n = 2$ and 3. The slight difference in the WPSNR value of the two algorithms is due to the embedding method used in the data embedding step. The embedding methods of the proposed algorithm provide a high embedding efficiency, therefore, the WPSNR of the proposed method is high while the method used by the n-LSBS algorithm does provide any embedding efficiency and therefore the WPSNR value is reduced. In terms of the percentage modification rate, the proposed algorithm for $n = 1$ introduces 9% fewer modifications than LSBS while the modifications for $n > 1$ are 6% less than n-LSBS. The overall performance of both algorithms is still higher than the technique [21] which highlights the significance of the role played by the proposed complexity identification step in providing the high visual quality of the stego image.

Finally, the visual quality of the stego image as generated by the proposed data hiding algorithm is measured in terms of SSIM. The SSIM values over varying n are presented in Table 5. From the table it can be seen that the proposed algorithm achieves a higher value of SSIM in comparison with [21] for all n. An average increase of 0.0005, 0.0021 and 0.0071 is achieved over [21] for $n = 1$, 2 and 3 respectively. An example of cover and stego images (Barbara and Cameraman) is presented in Fig. 12. The embedding is performed in the cover images using $\varepsilon = 1$. From the figure it is observed that both the cover and stego segments are visually similar and HVS cannot detect the embedding distortion.

For a comprehensive comparison, Table 6 presents the high visual quality performance of the proposed algorithm with respect to three techniques e.g., PVD (Pixel Value Difference) [31], TBPC (Tree-Based Parity Check) [24] and ATBPC (Adaptive TBPC) [24]. Each given image is evaluated with respect to the PSNR, WPSNRthe and SSIM at the embedding capacity EC of 30% and 50% for 1bpp case. It can be observed that a similar PSNR is achieved for the methods TBPC and ATBPC. Since both the techniques use a similar embedding approach of Tree-based parity check. The method ATBPC results in a higher WPSNR and SSIM as it is a content-adaptive approach. While the content-adaptive approach of the proposed technique performs better than ATBPC by achieving the highest WPSNR and SSIM at the given embedding capacity.

Fig. 12 shows the PSNR and WPSNR performance of the proposed method for multibit embedding on BOWS2 dataset. The comparison is made between three techniques e.g., adaptive PVD [33], Tri-PVD (TPVD) [34] and Edge-XOR embedding [23] for a variable embedding capacity of 10-70%. The performance of the proposed method is evaluated using the parameter $n = 3$ which employs the nbpp embedding upto 3 bits per pixel. It is clear from the figure that the proposed embedding algorithm produces highest quality stego images in comparison with the previous methods. The PSNR results in Fig. 8a shows that the proposed method is efficient in embedding by achieving PSNR value almost equal to that of Edge-XOR embedding method. On the other hand, the WPSNR results show that the proposed method achieves the highest visual quality by achieving highest WPSNR value for all embedding capacity. The reason for such high margin is that unlike previous approaches the proposed method utilizes the pixel difference in all directions and employs the most suitable difference for the calculation of embedding capacity per pixel. On the other hand, the technique of

TABLE 6. PSNR, WPSNR and SSIM comparison of proposed method with three techniques [24], [31] and [32].

Techniques		PVD [31]	TBPC [32]	ATBPC [24]	Proposed	PVD [31]	TBPC [32]	ATBPC [24]	Proposed
Images	EC (%)	30%				50%			
Lena	PSNR	54.25	57.42	57.28	58.72	52.51	55.34	55.35	56.49
	WPSNR	68.47	69.02	75.53	77.7	67.41	67.45	68	72.47
	SSIM	0.9994	0.9991	0.9994	0.9999	0.9982	0.9987	0.9988	0.9999
Baboon	PSNR	55.53	57.39	57.12	58.67	52.23	55.30	55.13	56.49
	WPSNR	94.85	81.93	92.12	106.14	86.58	79.55	81.07	84.82
	SSIM	0.9998	0.9997	0.9998	0.9999	0.9993	0.9995	0.9995	0.9999
Cameraman	PSNR	54.68	57.39	57.41	58.71	53.03	55.31	55.31	56.47
	WPSNR	69.28	62.45	65.76	92.64	66.90	65.32	66.01	63.92
	SSIM	0.9998	0.9987	0.9991	0.9999	0.9978	0.9983	0.9984	0.9999
Peppers	PSNR	54.24	57.42	57.33	64.01	52.49	55.39	55.38	56.51
	WPSNR	72.75	70.25	73.13	74.29	67.63	68.49	68.84	70.65
	SSIM	0.9992	0.9983	0.9994	0.9999	0.9984	0.9988	0.9988	0.9999
Peppers	PSNR	54.43	57.38	57.14	58.73	52.42	55.35	55.38	55.22
	WPSNR	68.86	70.31	83.72	106.53	66.82	68.42	69.31	76.54
	SSIM	0.9994	0.9995	0.9996	0.9999	0.9989	0.9991	0.9992	0.9999

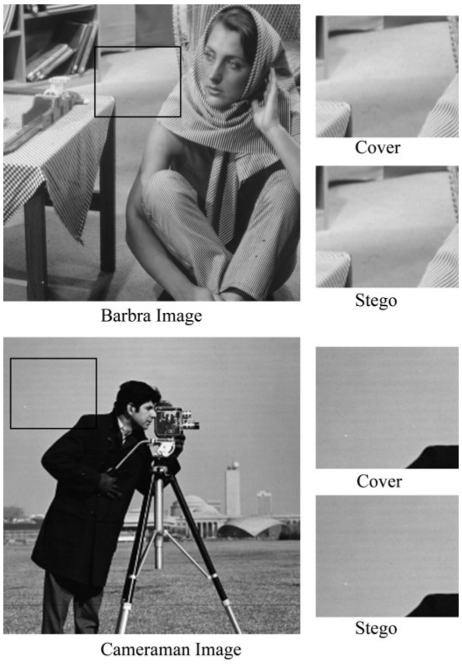


FIGURE 12. Subjective analysis of visual quality of stego images of the proposed content-adaptive image steganography.

Edge-XOR embedding calculates pixel difference in limited directions and then uses the average of the differences for the calculation of embedding capacity per pixel. The average of differences is not an appropriate way of calculating the data capacity per pixel since two blocks carrying varying intensity may have same average difference. In conclusion the results on BOWS2 dataset shows that the accuracy of the proposed complexity estimation method produces highest quality stego images by utilizing the highest complex pixels for embedding.

B. UNDETECTABILITY IN SPAM STEGANALYSIS DOMAIN

The subtractive pixel adjacency model is chosen for the measurement of detectability of the steganographic distortion. SPAM features reflect the pixel correlation in eight directions using a high pass filter in a small window and these noise residuals are then modeled as a higher-order Markov Chain. The resulting matrix of transition probabilities is utilized as a feature vector for the classifier. The steganographic distortion forces the cover properties to deviate from the SPAM model and thus the pixels where the correlation is high will be easily detected by this method. Given the cover and respective stego images from the BOSSbase image dataset, the 2nd order SPAM features of dimensions 686 are calculated and fed to an ensemble classifier. The data set is split into half training and the remaining half for testing dataset. The ensemble classifier consists of a batch of classifiers to which the feature set is randomly and equally distributed. A final decision is made based on the results from all the classifiers using majority voting. An out of bag error (OOBE) is the probability of false detection and is expected to be a high value for a secure steganographic method.

Table 7 shows the classification results as a measure of OOBE of the proposed method in comparison with the techniques EA-LSBMR [17] and MPBDH [35]. From the table, it can be seen that the proposed technique performs better than the two techniques. This shows that the proposed method accurately identifies the high texture regions in the cover image and the SPAM steganalysis thus lacks in predicting the embedding locations. The technique of MPBDH identifies the randomness in the bit plane of the cover image and uses

TABLE 7. The classification results in terms of the OOBE in the SPAM steganalysis domain.

Payload	OOBE			
	MPBDH [35]	EA-LSBMR [17]	Proposed	Improvement
10	0.3100	0.366	0.4743	21.66
20	0.236	0.338	0.3800	8.40
30	0.192	0.280	0.250	-6
40	0.111	0.216	0.240	0.9981

the block data hiding for embedding into each bit plane. However, the technique is not secure since the randomness in the bit plane of a cover image does not translate to a highly complex texture in the spatial domain. The edge adaptive technique [17] identifies the unidirectional edges and uses the LSB matching revisited technique for embedding, therefore the security performance is better as compared to MPBDH. The proposed technique, however, uses an eight directional approach for the texture identification of a region and therefore performs best among the two approaches.

VI. CONCLUSION

A content-adaptive steganography method is presented which aims to identify the complexity of the texture content of the image for data hiding. The complexity of the content is analyzed based on a high pass filter bank which computes pixel correlation in eight directions. In a local region, a high pass filter (HPF) bank is applied and eight residual responses are obtained. Following the proposed CBP criterion, a complexity level out of nine levels is assigned to an individualized pixel block. The pixels are then arranged in the priority of complexity from highest to lowest. Data embedding for the corresponding complexity level then takes place using proposed adaptive embedding algorithm. Experiments are performed for the analysis of embedding capacity, image quality and security on three datasets. Highest values of the IQ (image quality) parameters e.g., SSIM and WPSNR show the effectiveness of the proposed method.

In future the aim is to exploit the pixel variation to an increased pixel block size to further improve the texture analysis of the proposed method. We also aim to perform the security analysis of the proposed method using latest rich steganalysis models.

REFERENCES

[1] K. Bhaskar, P. Jerry, K. S. Ho, and S. Cyrus, "Applications, and services, 'I3: An IoT marketplace for smart communities,"' in *Proc. 16th Annu. Int. Conf. Mobile Sys.*, 2018, pp. 498–499.

[2] S. E. Bibri, "The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability," *Sustain. Cities Soc.*, vol. 38, pp. 230–253, Apr. 2018.

[3] R. Mehta, J. Sahni, and K. Khanna, "Internet of Things: Vision, applications and challenges," *Procedia Comput. Sci.*, vol. 132, pp. 1263–1269, 2018.

[4] S. R. Kim, J. N. Kim, S. T. Kim, S. Shin, and J. H. Yi, "Anti-reversible dynamic tamper detection scheme using distributed image steganography for IoT applications," *J. Supercomput.*, vol. 74, no. 9, pp. 4261–4280, Sep. 2018.

[5] R. Yagnik and K. Chetan, "Ubiquitous computing data security-a literature survey," *Int. J. Innov. Knowl. Concepts*, vol. 2, no. 2, pp. 122–127, Oct. 2018.

[6] L. Debina and T. Themrichon, "A survey on digital image steganography: Current trends and challenges," in *Proc. 3rd Int. Conf. Internet Things Connected Tech. (ICIoTCT)*, vol. 69, 2018, pp. 26–27.

[7] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, Mar. 2019.

[8] E. Toch, C. Bettini, E. Shmueli, L. Radaelli, A. Lanzi, D. Riboni, and B. Lepri, "The privacy implications of cyber security systems: A technological survey," *ACM Comput. Surv. (CSUR)*, vol. 51, no. 2, pp. 1–27, Feb. 2018.

[9] A. Mahmoud, R. Giovanni, and C. Bruno, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Secur. Appl.*, vol. 38, pp. 8–27, Feb. 2018.

[10] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. San Mateo, CA, USA: Morgan Kaufmann, 2007, pp. 322–330.

[11] D. Pooja, "Traditional and hybrid encryption techniques: A survey," in *Networking Communication and Data Knowledge Engineering*. Singapore: Springer, 2018, pp. 239–248.

[12] Y. Rai and P. Le Callet, "Visual attention, visual salience, and perceived interest in multimedia applications," in *Academic Press Library in Signal Processing*, vol. 6. Amsterdam, The Netherlands: Elsevier, 2018, pp. 113–161.

[13] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[14] X. Liao, J. Yin, S. Guo, X. Li, and A. K. Sangaiah, "Medical JPEG image steganography based on preserving inter-block dependencies," *Comput. Electr. Eng.*, vol. 67, pp. 320–329, Apr. 2018, doi: 10.1016/j.compeleceng.2017.08.020.

[15] K. Karampidis, E. Kavallieratou, and G. Papadourakis, "A review of image steganalysis techniques for digital forensics," *J. Inf. Secur. Appl.*, vol. 40, pp. 217–235, Jun. 2018.

[16] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. Ho, and K.-H. Jung, "Image steganography in spatial domain: A survey," *Signal Process., Image Commun.*, vol. 65, pp. 46–66, Jul. 2018.

[17] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 201–214, Jun. 2010.

[18] S. Islam, M. R. Modi, and P. Gupta, "Edge-based image steganography," *EURASIP J. Inf. Secur.*, vol. 2014, no. 1, p. 8, 2014.

[19] S. Chakraborty, A. S. Jalal, and C. Bhatnagar, "LSB based non blind predictive edge adaptive image steganography," *Multimedia Tools Appl.*, vol. 76, no. 6, pp. 7973–7987, Mar. 2017.

[20] G. Swain, "Very high capacity image steganography technique using quotient value differencing and LSB substitution," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 2995–3004, Apr. 2019.

[21] A. A. Mohammad, A. Al-Haj, and M. Farfoura, "An improved capacity data hiding technique based on image interpolation," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 7181–7205, Mar. 2019.

[22] G. Liu, W. Liu, Y. Dai, and L. S. , "Adaptive steganography based on block complexity and matrix embedding," *Multimedia Syst.*, vol. 20, no. 6, pp.227–238, 2014.

[23] H. Al-Dmour and A. Al-Ani, "A steganography embedding method based on edge identification and XOR coding," *Expert Syst. Appl.*, vol. 46, pp. 293–306, Mar. 2016.

[24] H. Al-Dmour, N. Ali, and A. Al-Ani, "An efficient hybrid steganography method based on edge adaptive and tree based parity check," in *Proc. Int. Conf. Multimedia Modeling*, 2015, pp. 1–12.

[25] G. Paul, I. Davidson, I. Mukherjee, and S. S. Ravi, "Keyless dynamic optimal multi-bit image steganography using energetic pixels," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 7445–7471, Mar. 2017.

[26] Z. Yuqian, L. Ding, and H. Thomas, "A new rule for cost reassignment in adaptive steganography," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2654–2667, Nov. 2017.

[27] *SIPI (Signal and Image Processing Institute) Image Dataset*. Accessed: 1997. [Online]. Available: http://sipi.usc.edu/database/database.php?volume=misc&image=1#top

[28] P. Bas and T. Furon. (2007). *BOWS-2*. [Online]. Available: http://bows2.ec-lille.fr/

[29] P. B. O. Pevny and T. Filler. (2013). *Break Our Steganography System*. [Online]. Available: http://agents.fel.cvut.cz/boss/index.php?mode=VIEW&tmpl=materials

[30] S. Kaur, H. Kour, and D. Sen, "Image quality measurement through structural similarity based on higher order moments," in *Proc. IEEE Annu. India Conf. (INDICON)*, Dec. 2016, pp. 1–6.

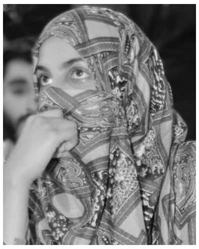
[31] S. Kaur, H. Kour, and D. Sen, "Data hiding in gray-scale images using pixel value differencing," in *Technology Systems and Management*. Berlin, Germany: Springer, 2011, pp. 27–33.

[32] R. Y. Li, O. C. Au, K. K. Lai, C. K. Yuk, and S.-Y. Lam, "Data hiding with tree based parity check," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2007, pp. 635–638.

[33] J. K. Mandal and D. Debashis, "Steganography using adaptive pixel value differencing (APVD) of gray images through exclusion of overflow/underflow," in *Proc. 2nd Int. Conf. Comput. Sci., Eng. Appl. (CCSEA)*, 2012, pp. 93–102.

[34] Y.-P. Lee, J.-C. Lee, W.-K. Chen, K.-C. Chang, I.-J. Su, and C.-P. Chang, "High-payload image hiding with quality recovery using tri-way pixel-value differencing," *Inf. Sci.*, vol. 191, pp. 214–225, May 2012.

[35] T. D. Nguyen, S. Arch-int, and N. Arch-int, "An adaptive multi bit-plane image steganography using block data-hiding," *Multimedia Tools Appl.*, vol. 75, no. 14, pp. 8319–8345, Jul. 2016.



AYESHA SAEED received the B.Sc. degree in telecommunication from the University of Engineering and Technology Taxila, Pakistan, in 2017, where she is currently pursuing the M.Sc. thesis in telecommunication engineering. Her research interests include information security systems, computer vision, pattern recognition, and image processing. She joined Association for Computing Machinery (ACM) society, in 2016 and played active part as a Finance Secretary and a General Member. She has actively organized many technical events in university campus.

FAWAD (Student Member, IEEE) received the B.S. degree in telecommunication from the National University of Modern Languages, Pakistan, in 2011, and the M.S. degree from the Military College of Signals, National University of Science and Technology, Pakistan, in 2016, which was followed by his Graduate Assistant job in Ghulam Ishaq Khan Institute of Engineering Sciences and Technology. He is currently pursuing the Ph.D. thesis in telecommunication engineering from the University of Engineering and Technology Taxila, Pakistan. In 2012, he joined Zong China mobile company as a Transmission Engineer. His research interests include computer vision, pattern recognition, and image processing.

MUHAMMAD JAMIL KHAN (Member, IEEE) received the B.Sc. Engineering degree in computer engineering, the M.Sc. degree in telecommunication engineering, and the Ph.D. degree in computer engineering from University of Engineering and Technology Taxila, Taxila, Pakistan, in 2005, 2009, and 2016 respectively. He is currently an Assistant Professor and the Director of Embedded Systems and Digital Signal Processing Laboratory, University of Engineering and Technology Taxila, where he is also the founder of Virtual Reality Simulation Laboratory. He has authored or coauthored numerous technical articles in well-known international journals and conferences. His current research interests include multimedia content analysis, RF identification, and machine learning.

HUMAYUN SHAHID (Member, IEEE) received the B.S. degree in communication systems engineering from the Institute of Space Technology, Islamabad, in 2008, and the M.S. degree in signal processing from Nanyang Technological University, Singapore, in 2011. He joined Space and Upper Atmosphere Research Commission (SUPARCO), where he worked on radiation-hardened space grade components for telemetry subsystems. Thereafter, he joined the Department of Telecommunication Engineering, University of Engineering and Technology Taxila, Taxila, where he currently works as an Assistant Professor. He is affiliated with the ACTSENA research group working towards design and signal processing-related aspects of electromagnetic transduction-based sensor-incorporated chipless RFID tags. He is also the Director of the Departmental Antenna and RF Laboratory. His research work has been featured in a number of ISI-indexed journals and international conferences.

SYEDA IFFAT NAQVI received the B.Sc. Engineering degree in computer engineering and the M.Sc. degree in telecommunication engineering from the University of Engineering and Technology Taxila, Taxila, Pakistan, in 2006 and 2011, respectively. She is currently an Assistant Professor and also associated with the ACTSENA Research Group, University of Engineering and Technology Taxila. She is working towards the design and implementation of multiple antenna array systems for current 4G and next generation millimeter-wave 5G mobile communication applications. She has authored or coauthored numerous technical articles in ISI-indexed journals and international conferences.

MUHAMMAD ALI RIAZ (Member, IEEE) received the M.S. and B.S. degrees in electrical engineering from Iowa State University, USA, in 2010 and 2009, respectively. Afterwards, he joined the Department of Electrical and Computer Engineering, Iowa State University, USA, as a Research Assistant. He is currently serving as an Assistant Professor associated with ACTSENA Research Group, University of Engineering and Technology Taxila, Taxila. He is working towards the design and implementation of chipless RFID tags based on electromagnetic signature and their signal processing applications. He also serves as the Director of the Electronics and Measurements Laboratory at his department. His research work has been featured in a number of ISI-indexed journals.

MANSOOR SHAUKAT KHAN received the B.Sc. degree from PU Lahore, Pakistan, in 1994, the M.Sc. degree in statistics from the University of ARID Agriculture Rawalpindi, Pakistan, in 2000, and the Ph.D. degree from the Beijing Institute of Technology (BIT), China, in January 2016. He is currently an Assistant Professor with the Department of mathematics, COMSATS University Islamabad, Pakistan. His research interests include mathematical modeling and optimization, quality and reliability engineering, survival analysis, spatial data analysis, and data science. He has more than ten research articles in reputed journals and proceedings of international conferences. Most recently, he received National Research Program for Universities award from HEC Pakistan.

YASAR AMIN (Senior Member, IEEE) received the B.Sc. degree in electrical engineering with specialization in telecommunication and the M.B.A. degree in innovation and growth from the Turku School of Economics, University of Turku, Finland, and the M.Sc. degree in electrical engineering with specialization in system on chip design and the Ph.D. degree in electronic and computer systems from the Royal Institute of Technology (KTH), Sweden, with the research focus on printable green RFID antennas for embedded sensors. He is currently an Associate Professor and the Chairman of the Telecommunication Engineering Department, University of Engineering and Technology Taxila, Pakistan. He also serves as the Director of Embedded Systems Research and Development Centre. He is the Founder of Agile Creative Technologies for Smart Electromagnetic Novel Applications (ACTSENA) Research Group. He has authored or coauthored more than 100 international technical articles in conferences and journals. His research interests include the design and application of multiple antenna systems for next generation mobile communication systems, millimeter-wave and terahertz antenna array, implantable & wearable electronics, and inkjet printing technology in microwave applications. He is a member of more than a dozen international professional societies and the Fellow of PAE.