


Improving data hiding within colour images using hue component of HSV colour space

Fatuma Saeid Hassan | Adnan Gutub 

Computer Engineering Department, Umm Al-Qura University, Makkah, Saudi Arabia

Correspondence

Adnan Gutub, Computer Engineering Department, Umm Al-Qura University, Makkah, Saudi Arabia.
Email: aagutub@uqu.edu.sa

Abstract

Data hiding technologies aim to hide the existence of secret information within digital covers such as images by causing unnoticeable degradation to their quality. Reducing the image distortion and increasing the embedding capacity are the main points that the data hiding techniques revolved around. This article proposes two high payload embedding methods with high stego image quality using the Hue-Saturation-Value (HSV) colour model. The first method is hue-based embedding (HBE) that employs the H plane for hiding one or two bits in non-grey pixels. The second method uses the three HSV components, so it is called three-planes embedding (TPE). In TPE, one bit is hidden in the least significant bit (LSB) of V of the grey pixels, one or two bits in H of the pixels having low saturation or low brightness and one bit in the LSB of S otherwise. The experiments were conducted on 25 images and the results show that HBE hides more data on average than TPE with its quality reaching 60 dB. TPE achieves quality up to 61 dB and capacity reaches 364 Kb. TPE scores the highest capacity among six state-of-the-art techniques in Red-Green-Blue, HSV, Hue-Saturation-Intensity and YCbCr spaces with the highest average peak signal to noise ratio midst five of them. By embedding 60, 90, and 120 Kb, this TPE attains the best average quality amid all the methods.

1 | INTRODUCTION

With the widespread use of the Internet, sensitive contents have become more exposed to penetration by intruders. Therefore, the data hiding technique, or as it is known as steganography [1], involves covert communication between the authorised parties who transmit the secret information. This process needs an embedding algorithm and a host object, like image [2], video [3] or text [4, 5], as its basic components to conceal the secret. Data hiding methods can be reversible or irreversible. Reversibility indicates the possibility of reshaping the exact carrier object once the message is retrieved, while irreversibility does not guarantee so [6]. In either of the two types, the human visual system must be taken into consideration in order to not attract the attackers attention to the presence of secret data within the host media. This is because the better the imperceptibility property of the steganography method, the better its security. Additionally, the more amount

of payload that the method can hide, the higher its capacity. Research studies that address the embedding within images use either the grayscale [7–9] or colour images. A hiding technique using grey images could be applied for colour images [10] but by repeating it in the three planes of the images independently [11], which means increasing the time-complexity. A better way for embedding into colour images is by using the techniques that are dedicated to this type of images. Colour-image steganography methods in the literature use different colour spaces such as Red-Green-Blue (RGB) [11–14], Hue-Saturation-Value (HSV) [15, 16], Hue-Saturation-Lightness (HSL) [17], Hue-Saturation-Intensity (HSI) [18, 19] and YCbCr [20, 21] that stands for luminance, chroma Blue difference and chroma Red difference.

RGB is the basic model in most display devices [22] in which the colour is represented as a combination of three primary colours: Red, Green and Blue. Mixing two full primary colours produces a secondary colour, so Red-Green is Yellow,

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

Green-Blue is Cyan and Blue-Red is Magenta. Additionally, mixing one full primary colour with one secondary colour yields a tertiary colour, so there are six colours of this type. RGB is a correlated model because a small modification in one of its components influences the images quality [15]; so it is not the most suitable model for data hiding. Therefore, there is a need for studying the hiding using the other colour spaces, which are uncorrelated colour spaces, for the sake of improving the performance. Conversion between the RGB and the three hue-based colour models was given in [23]. Also, converting RGB to YCbCr and vice versa was given in [20].

In each of the HSV, HSL and HSI colour spaces, hue is the colour's name, saturation is the colour's purity and the third component (value, lightness or intensity) is the colour's brightness. The H plane has values between 0° and 360° where 0° is the same as 360° . The S and I planes are percentage values from 0% to 100%. When S is 0%, the grayscale values appear by varying the values of the V plane from 0% (Black) to 100% (White). In this case, the hue is undefined. Additionally, if the colour is Black then both H and S are undefined. Only in the HSL model, H and S are also undefined if the colour is White. A close look at the hue plane shows that the primary, secondary and tertiary colours of the RGB model are separated by 30° in the H plane. Thus, they produce 12 basic hues that are Red (at 0° or 360°), Red-Yellow (at 30°), Yellow (at 60°), Yellow-Green (at 90°), Green (at 120°), Green-Cyan (at 150°), Cyan (at 180°), Cyan-Blue (at 210°), Blue (at 240°), Blue-Magenta (at 270°), Magenta (at 300°) and Magenta-Red (at 330°). Therefore, an alteration in the value of H may change the perceived colour significantly [24], which makes the hue plane the least suitable plane for data embedding [17]. On the other hand, a careful division of the hue values into clusters for data hiding decreases the chance of detecting the change by human eyes, as we propose in this article.

In the current work, we propose dual irreversible data hiding methods that utilise the hue plane of the HSV colour space for carrying data. The reason for choosing the HSV model among the HSL and HSI models is its efficiency [16] and its suitability for our data embedding methods, as proven by our experiments. We divide the H plane into groups depending on the 12 basic hues in order to maintain the quality of the images while increasing the hiding capacity. The first method uses only the hue component while the second one employs all the three planes of the HSV colour space for information concealing.

The rest of the article is organised as follows: The following section discusses a literature review of the existing irreversible data hiding techniques. A detailed description of the two proposed methods is given in Section 3. Section 4 analyses the experimental results of the methods. Finally, the conclusion of the article is given in Section 5.

2 | LITERATURE REVIEW

In this section, we provide a brief discussion on some of the state-of-the-art techniques in the field of data hiding within colour images. The most common method is the least

significant bit (LSB) substitution in which the LSB of the binary representation of each pixel in the image is replaced by a secret bit [14]. The method can be stretched to increase the embedding capability by replacing more LSBs of the pixels, but this leads to more distortion in the image [14, 25]. Karim et al. [12] used the simple LSB substitution method to embed data in either the Green or Blue components of the pixels. In their method, the result of the XOR operation between one bit of a secret key and the LSB of the Red component determined the embedding plane in the pixel. This dependency on the secret key and the change in the embedding position improved the security of the simple LSB technique. The concept of Karim et al.'s method [12] was adopted in [14] but with adding several security barriers that involved permuting the image pixels, encrypting the secret key and encrypting the secret data. This method was not only highly secure but also computationally inexpensive. Muhammad et al. [18] investigated the utilisation of the LSB method for hiding data in the intensity channel of the HSI colour space. The reported results of their method showed more improvement in the image quality than Karim et al. [12]. Another work that utilised the I plane and improved the imperceptibility of Karim et al.'s method was nominated by the authors in [19]. Their work encoded the secret message using a three-level encryption algorithm. Then the image was scanned in a Morton manner to embed the cipher text by LSB substitution. To use the LSB method in the HSV colour space, the proposed technique in [15] divided the value plane into equal-sized blocks to embed the encrypted secret data into the positions specified by the magic matrix. We refer to this technique in this article as the HSV-MLSB. This technique achieved higher average visual quality than [18] when hiding the same amount of data in multiple cover images. The visual imperceptibility of the technique was better than the method in [25], which used the specific I channel of the HSI model, that is, to hide the secret content with the help of the magic matrix, a proposed multi-level encryption algorithm and the LSB method were used. More image quality enhancement of the two methods in [18, 25] was achieved by the proposed technique in [16] that encrypted the sensitive data using an iterative magic matrix encryption algorithm. Then, the data was embedded in a scrambled version of the V plane of the HSV space using an adaptive LSB method. Three different keys were used in the method that made it very secure but increased the key exchange overhead. Kumar et al. [26] presented a selective embedding technique in the RGB images in which two conditions must be satisfied to utilise a pixel for embedding: the three components of the pixel have different values and the least component is less than the others by two values at minimum. The least component of each pixel that fulfilled the criteria was employed for concealing one secret bit using the LSB method. This technique produced very little distortion in the images. However, it could embed only a small size of payload due to the strict embedding criteria. Alwan et al. [20] proposed an LSB-based technique that used the YCbCr colour model in a way that improved the security of the basic LSB method. Their idea was to apply the XOR operation between one secret bit and the achromatic component of the pixel.

Then, the result was again XORed with the C_b component to yield the bit that was to be finally embedded into the LSB of the C_r channel. The reported peak signal to noise ratio (PSNR) values of the method were more than 54 dB for all the images, which indicates the high quality of the images.

Gutub [13] presented a simple, secure and high-capacity data hiding method in which one of the RGB channels of a pixel was used to indicate the capability of the other two channels, which were called data holders, to embed two secret bits using LSB substitution. The first indicator channel was selected depending on the size of the secret message and the subsequent indicator channels were selected sequentially. Swain and Lenka [27] used a fixed indicator channel which was the maximum value summed for all the pixels. Their method encrypted the secret bits to add a second security layer to the pixel indicator technique in [13]. Then, it hid four bits at maximum in one of the two data channels depending on predefined conditions. The authors reported a high PSNR value of 42.75 dB as the minimum visual quality of their high payload method. In Das and Kars method [28], the indicator channel was also used as a data holder and its information was embedded along with the encrypted secret data to be used in the extraction phase. Their method embedded two bits in the indicator channel and three bits in each of the other two channels, which increased the hiding capacity of [13]. In Pandey et al.'s method [29], the indicator channel might also have been used for hiding data so that the embedding rate could reach 6 bits per pixel. Their method improved the embedding rate of Gutub's method [13], with high visual quality ranging from 48 to 71 dB when hiding 512 bytes of data.

Ioannidou et al. [30] took advantage of the idea that embedding in the edge pixels is more efficient than in the non-edge pixels in terms of the payload capacity and the visual quality. In comparison, the improvement in [31] proposed multibit assignment steganography for palette images, in which each gregarious colour that possesses close neighbouring colours in the palette is exploited to represent several secret bits, that is, replacing the original colour with a selected neighbouring colour in a smart way. Although the work was complex and challenging, it showed impressive payload-distortion performance. In comparison, Grover and Mohapatra [32] proposed an edge-based technique of high imperceptibility to be used for colour-image authentication. Starting from the centre pixel in the B channel, the method embedded two secret bits in the smooth pixels and three bits in the sharp pixels using the LSB method. Rashid and Majeed [33] proposed a technique that applied edge detection on the Blue component and used the edge positions as indicators of the existence of data in the Green component. This method used only the edge positions that satisfied a threshold value, which increased the security of the method.

Some of the modern steganographic methods defined a distortion function to minimise the embedding detected alteration [10, 11, 34, 35]. For example, Yaofei Wang and his team in [10] discussed image steganography utilising the differences between the colour channels. Interestingly, they

observed the G channel which had specifically stronger correlation than others, that is, its resistance to detection was better. Furthermore, the work in [10] innovated non-additive costs for colour image steganography, distributing the embedding capacity between the three channels in a smart way so as to not violate the Complexity Prior rule. Similarly, the payload partition image steganography in [11] relied on amplifying channel modification probabilities to assign embedding capacity to RGB channels. Their clustered strategy made the embedding changes concentrated in textured regions, achieving better resistance to many steganalysis. Some distortion functions were additive in which the costs were summed over to form the function (clarified in [11]) and some were non-additive as in [10] as well as in [34, 35].

The techniques mentioned so far worked in the spatial domain in which the image pixels were manipulated directly to embed the secret content. The techniques that were designated to work in the frequency domain transformed the pixels into the frequency distribution. Then the produced coefficients were utilised for embedding. In the end, the altered coefficients were inversely transformed to form the stego image. Some examples of the transforms used for data hiding are the discrete Fourier transform [21], discrete cosine transform [36] and the discrete wavelet transform [37]. These types of techniques are computationally complex and can carry fewer data [16] compared with the spatial-domain techniques.

In this article, we propose two methods that work in the spatial domain and their embedding performance is the best among six LSB-based techniques, with a high visual quality of the produced stego images. The novel contributions of this article are briefed below:

- Splitting the 360 hue values of the HSV colour model into 24 groups of equal size and assigning a specific hue name for each group. The names are derived from the 12 basic hues. Each group is further divided into clusters of two or three members. We make the embedding in the hue plane of a pixel to be by changing the hue value to another value but within the same cluster and group. In this way, we improve the image quality of the embedding in the hue plane.
- Improving the embedding capacity of six existing 1LSB-based methods by proposing two data hiding techniques.
- Improving the average visual quality of three existing methods by proposing the hue-based embedding (HBE) data hiding technique. It uses the hue of only the saturated pixels to embed one or two secret bits.
- Improving the average visual quality of five existing methods and the proposed HBE method by proposing the three-planes embedding (TPE) data hiding technique. It uses one of the three HSV planes for embedding in each pixel. The embedding plane in a pixel is specified depending on the value of the S and V planes. When the embedding plane is H , two bits at maximum are embedded using the HBE method. Otherwise, one bit is embedded using the simple LSB substitution method.

3 | PROPOSED METHODS

This work presents two improvements. The general processes that are involved in the embedding and extraction phases are illustrated in Figure 1. At first, the cover image is converted from the RGB to the HSV model to produce the HSV image. Then, one of the two proposed embedding methods is applied to hide the secret message within the resultant image. The output of this process is the image in the HSV space that encloses the sensitive data. This image is then converted back to the RGB model in order to form the final stego image. For data extraction, the stego image is converted to HSV and then the corresponding extraction technique is applied to the image.

In Section 3.1, we discuss the first proposed embedding method followed by discussing the corresponding extraction method in Section 3.2. The second proposed embedding technique is explained in Section 3.3 and the recovery process is described in Section 3.4.

3.1 | Data embedding using HBE method

In this section, we discuss the embedding phase of our first proposed technique, that is, the HBE method. It uses the H plane of the HSV colour space as the main component for hiding. The first step of this method is the conversion of the image from the RGB model to the HSV model by normalising the R , G and B components to be in the range $[0, 1]$ instead of $[0, 255]$. The next step is the calculation of the H , S and V components with the help of the equations provided in [38]. Then, the value of H is transformed from the range $[0, 1]$ to the range $[0, 360]$ by multiplying it by 360 and rounded to the nearest integer value. In this method, we hide one or two secret bits in the hue of the pixels that have saturation value not equal to 0%. On the other hand, there is no embedding in the other pixels when S equals to 0%.

This technique depends on categorising the hues of the HSV colour space based on the 12 basic hues. For simplicity, we give names for the tertiary hues as followed: Red-Yellow is Orange, Yellow-Green is Chartreuse green, Green-Cyan is Spring green, Cyan-Blue is Azure, Blue-Magenta is Violet and Magenta-Red is Rose. We further divide the hues into 24 groups and each group has a range of 15° . The names of the groups are taken from the basic 12 hues and the hues between each of them. These groups are represented as pies of hues in Figure 2. The green degrees marked in the figure represent the exact positions of the 12 basic hues. The following are a detailed view of the ranges and names used for the groups:

Red = $353^\circ\text{--}0^\circ$ ($360^\circ\text{--}7^\circ$)	Green = $113^\circ\text{--}120^\circ\text{--}127^\circ$	Blue = $233^\circ\text{--}240^\circ\text{--}247^\circ$
Red-Orange = $8^\circ\text{--}22^\circ$	Green-Spring green = $128^\circ\text{--}142^\circ$	Blue-Violet = $248^\circ\text{--}262^\circ$
Orange = $23^\circ\text{--}30^\circ\text{--}37^\circ$	Spring green = $143^\circ\text{--}150^\circ\text{--}157^\circ$	Violet = $263^\circ\text{--}270^\circ\text{--}277^\circ$
Orange-Yellow = $38^\circ\text{--}52^\circ$	Spring green-Cyan = $158^\circ\text{--}172^\circ$	Violet-Magenta = $278^\circ\text{--}292^\circ$

Yellow = $53^\circ\text{--}60^\circ\text{--}67^\circ$	Cyan = $173^\circ\text{--}180^\circ\text{--}187^\circ$	Magenta = $293^\circ\text{--}300^\circ\text{--}307^\circ$
Yellow-Chartreuse green = $68^\circ\text{--}82^\circ$	Cyan-Azure = $188^\circ\text{--}202^\circ$	Magenta-Rose = $308^\circ\text{--}322^\circ$
Chartreuse green = $83^\circ\text{--}90^\circ\text{--}97^\circ$	Azure = $203^\circ\text{--}210^\circ\text{--}217^\circ$	Rose = $323^\circ\text{--}330^\circ\text{--}337^\circ$
Chartreuse green-Green = $98^\circ\text{--}112^\circ$	Azure-Blue = $218^\circ\text{--}232^\circ$	Rose-Red = $338^\circ\text{--}352^\circ$

We further divide each group into clusters that have three members as follows: (h_1, h_2, h_3) , (h_4, h_5, h_6) , (h_7, h_8, h_9) , (h_{10}, h_{11}, h_{12}) and (h_{13}, h_{14}, h_{15}) . The Red group is a special case because it is not a continuous group as it is separated at 0° (360°). Therefore, its clusters may have three or two members as follows: $(353^\circ, 354^\circ, 355^\circ)$, $(356^\circ, 357^\circ)$, $(358^\circ, 359^\circ)$, $(0^\circ, 1^\circ, 2^\circ)$, $(3^\circ, 4^\circ, 5^\circ)$, $(6^\circ, 7^\circ)$.

After obtaining the HSV image, the embedding process starts by scanning the image from left to right and top to bottom to embed the bits sequentially. For each pixel, we check if the value of S is greater than zero. If so, we embed data in the rounded H of this pixel. The embedding is done by first determining the group and the exact cluster within the group that this hue belongs to. Depending on the number of members in the cluster and the value of one secret bit, the number of bits n that can be embedded in the pixel and the new hues value H_{steg} are determined.

If the cluster has only two members, we embed one bit, so $n = 1$. In this case, the H_{steg} will be the first hue in the cluster if the bit is 0 while the H_{steg} will be the second hue in the cluster if the bit is 1. If the cluster is of three, we check the value of one secret bit; if it is 0, $n = 1$ and the H_{steg} will be the first hue in the cluster. If it is 1, $n = 2$ and the H_{steg} is determined as follows: if the next secret bit is 0, H_{steg} will be the second hue in the cluster; if the next bit is 1, the H_{steg} will be the third hue in the cluster.

It is worth to mention that the 360° belongs to the Red cluster ($358^\circ, 359^\circ$), but the cluster is still treated as a two-member cluster because 360° is the same as 0° . If we treat the cluster as a three-member cluster, there will be incorrect recovery of the secret bits in the data extraction phase. The main goals of the clusters used are to increase the embedding space while keeping the embedding value within the group and to reduce the difference between the old and the new values of the pixels hue to two at maximum.

The final steps in the embedding phase are the normalisation of the H to be back in the range $[0, 1]$, by dividing it by 360, and then the conversion of the image from HSV to RGB with the help of the equations provided in [38], to be sent to the receiver. Note that the RGB image is in the range $[0, 1]$.

A practical example of the embedding in the proposed HBE technique is represented in Figure 3. Using a cover image of size 2×2 , the process is started by converting the cover image to the HSV colour space and then scanning the pixels in

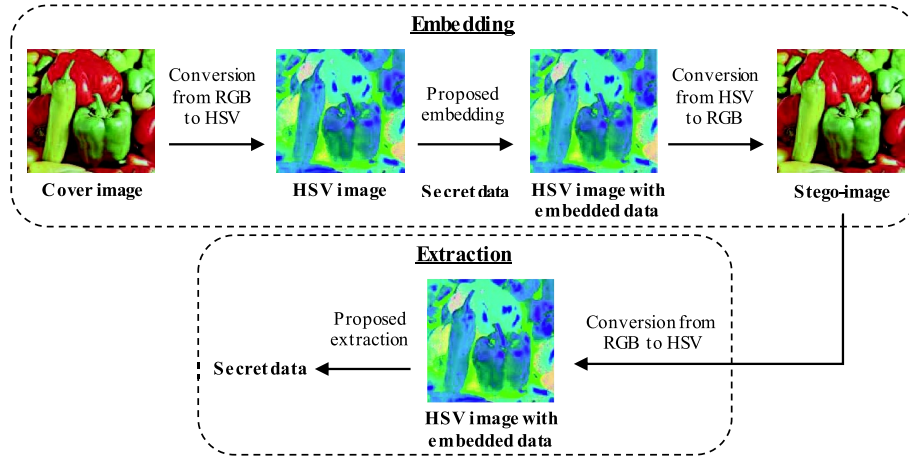


FIGURE 1 The general framework of the proposed scheme. HSV, Hue-Saturation-Value; RGB, Red-Green-Blue

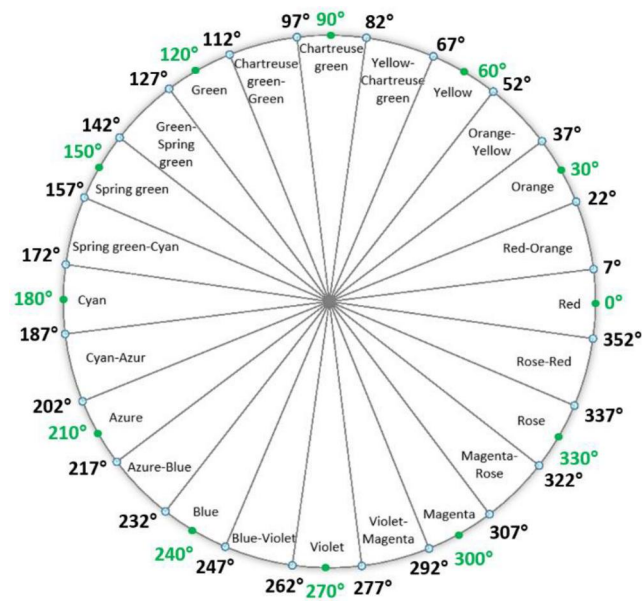


FIGURE 2 Geometric representation of the hue groups used in our hue-based embedding method

the following sequence: A, B, C and D. Assuming that the secret bit stream is (011,010), the embedding procedure is as follows:

For each pixel, we check its suitability for HBE data hiding by ensuring that its saturation value is not 0. Starting from pixel A, its hue is normalised to be in the range $[0, 360]$ and rounded to an integer. Then, we define its group as *Red-Orange* because the H falls in the range $(8^\circ-22^\circ)$. The exact cluster that it belongs to is $(14^\circ, 15^\circ, 16^\circ)$, which has three members; so we will check the value of one secret bit if it is 0 or 1. Because the bit is 0, we will embed one bit in the pixel, $n = 1$, and there is no need to check the next secret bit. Additionally, the stego value of H will be the first hue in the cluster, $H_{steg} = 14$. For pixel B, its H belongs to the group *Rose* and cluster $(323^\circ, 324^\circ, 325^\circ)$. It is a three-member cluster, and so one secret bit is checked and its value is 1. Thus, another bit

will be checked, which means that $n = 2$. The next bit is 1; therefore, the H_{steg} will be the third hue in the cluster, 325. Pixel C has a rounded hue of 360 and so it follows the *Red* group and the two-member cluster $(358^\circ, 359^\circ)$. In this type of cluster, we check only one secret bit and here it is 0; so the H_{steg} is the first hue in the cluster. Following the same procedure, the hue of pixel D belongs to the *Rose-Red* group and the three-member cluster $(347^\circ, 348^\circ, 349^\circ)$. One bit from the secret is 1, and so we will take another bit which is 0. Thus, the new hue value is the second hue in the cluster, $H_{steg} = 348$.

Next, all the hues are normalised back to the $[0, 1]$ range and combined with the original S and V planes to reshape the HSV image that contains the hidden data. Finally, the reconversion from HSV to RGB is applied to produce the stego image, which is in the range $[0, 1]$.

3.2 | Data extraction using the HBE method

On the receiver side, the secret message can be extracted from the stego image by applying the extraction procedure of the proposed HBE technique. The process is started by converting the image from the RGB space to the HSV space. Then, the H plane is normalised to be in the range $[0, 360]$ and rounded to an integer. The image is scanned sequentially to extract the bits from the pixels that have saturation value greater than zero. The following is the details of the extraction process:

For each pixel, we determine the group and the cluster that its hue belongs to. If the cluster is of two members, there is only one hidden bit in the pixel and it is determined by the value of the H_{steg} as follows: If the H_{steg} is the first hue in the cluster, the bit is 0; otherwise, the bit is 1. If the cluster contains three members, three cases are considered as follows: if the H_{steg} is the first hue in the cluster, there is only one bit hidden and it is 0; otherwise, there are two hidden bits in the pixel. These two bits are (10) if the H_{steg} is the second hue in the cluster and (11) if the H_{steg} is the third hue in the cluster. Finally, all the secret bits are concatenated together to form the entire secret stream.

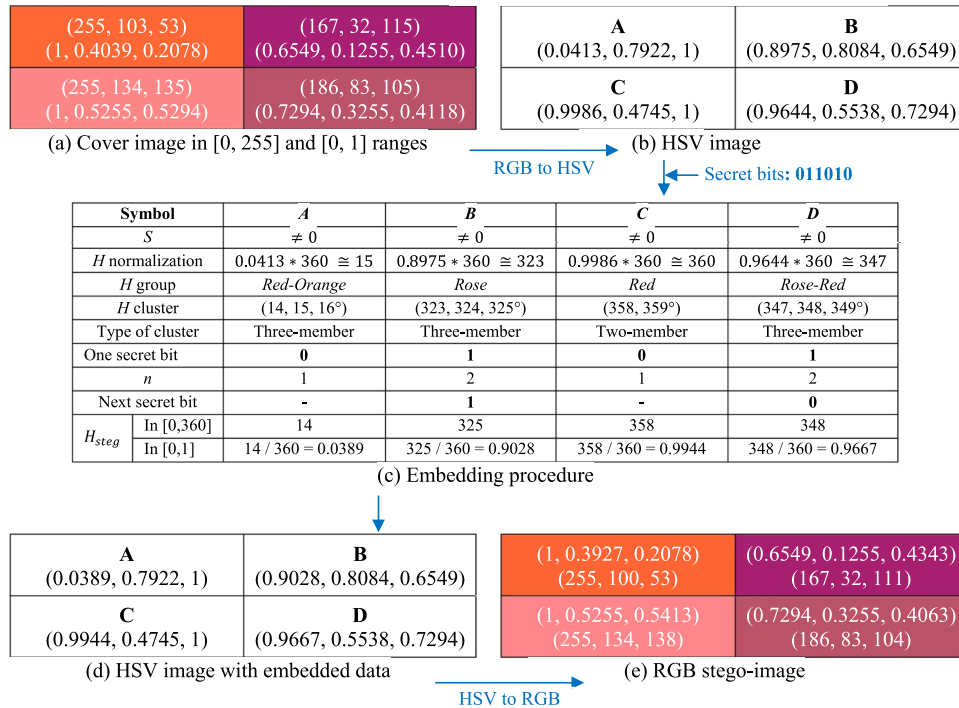


FIGURE 3 Example of embedding data into a 2×2 image using the HBE method. (a) Cover image in $[0, 255]$ and $[0, 1]$ ranges; (b) Hue-Saturation-Value (HSV) image; (c) Embedding procedure; (d) HSV image with embedded data; (e) Red-Green-Blue (RGB) stego image

Using the embedding example in Figure 3, we can extract the secret data by applying the above procedure. At first, the stego image is transformed to the HSV colour model and the output is the HSV image that contains the secret bits.

The normalised H values of all the four pixels are checked for data extraction because all of them are saturated. The cluster of pixel A is $(14^\circ, 15^\circ, 16^\circ)$ and the H_{steg} is the first hue in the cluster; so the secret bit is 0. Pixel B belongs to the $(323^\circ, 324^\circ, 325^\circ)$ cluster and its H_{steg} is the third pixel in the cluster, so two secret bits are extracted which are (11). The third pixel follows the $(358^\circ, 359^\circ)$ cluster and the stego hue is the first hue in the cluster; thus the secret bit is 0. The cluster of pixel D is $(347^\circ, 348^\circ, 349^\circ)$ and the H_{steg} is equal to the second hue in the cluster; so the two secret bits are (10). Finally, all the bits are concatenated as (011, 010) which is the full secret message.

3.3 | Data embedding using the TPE method

This section explains the hiding process of our second proposed technique that is called the three-plane embedding method. TPE uses the three planes of the HSV colour model for embedding. First, the cover image is converted to the HSV space using the method in [38]. Then, the H plane is normalised to the range $[0, 360]$ and the S and V components are normalised to the range $[0, 255]$. In TPE, all of the pixels are involved in the hiding process as follows:

For each pixel, we check whether it is unsaturated, that is, $S = 0$ or not. If so, one bit is embedded in the V plane using the simple LSB substitution method. If not, we check the values of the S and V planes. If the embedding condition ($|S \leq 51|$ or $|V \leq 51|$) is satisfied, the hiding takes place in the H plane using the HBE method; otherwise, the S plane is used for hiding one secret bit using the LSB method.

In other words, the V component is utilised for embedding in the grey pixels and the H component is utilised in the pixels that are either maximum 20% saturated or maximum 20% illuminated; otherwise the S component is utilised in the pixels that have both saturation and value above 20%. Note that 20% is equal to 51 in the range $[0, 255]$.

Finally, the three planes are normalised back to the range $[0, 1]$ and the image is retransformed to the RGB space using the conversion method in [38]. A numerical example of the TPE embedding process is given in Figure 4 and it is explained as follows:

Using a 2×2 cover image, it is converted to the HSV space and then the pixels are scanned from A to D to embed the bit stream (10,010). Starting from pixel A, the S plane is equal to zero which means it is an unsaturated pixel. So, the embedding will take place in the V plane on replacing the LSB of the normalised V by one secret bit. The other three pixels are saturated pixels; so the embedding condition is checked to decide whether the embedding plane is H or S . Pixel B does not satisfy the condition because its S and V are both above 20%; so the S plane will be the place for embedding one secret bit by the LSB substitution. Pixel C has a saturation of less than 20% and pixel D has a value of less than 20%; so both of them

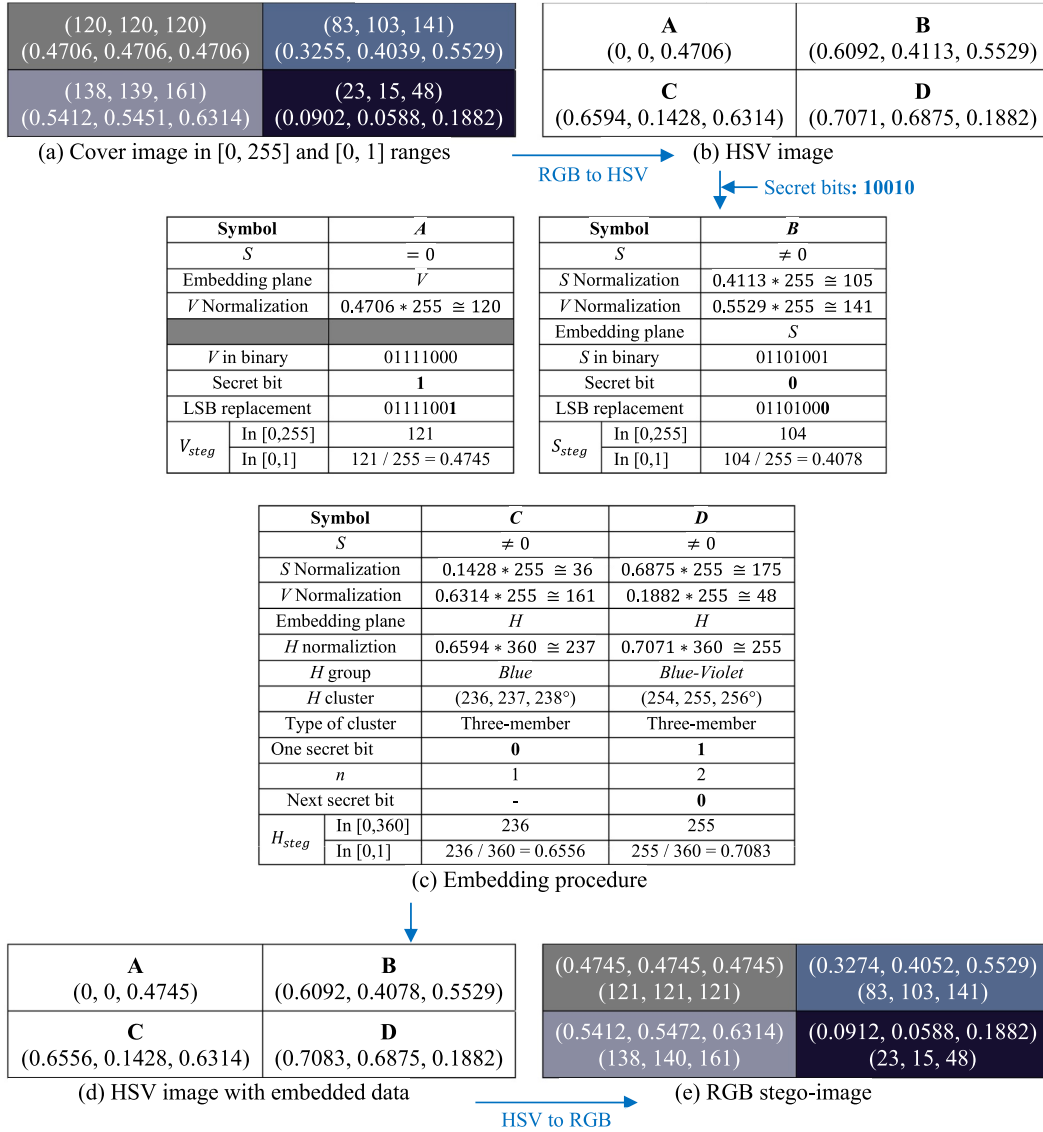


FIGURE 4 Example of embedding data into a 2×2 image using the three-planes embedding method. (a) Cover image in $[0, 255]$ and $[0, 1]$ ranges; (b) Hue-Saturation-Value (HSV) image; (c) Embedding procedure; (d) HSV image with embedded data; (e) Red-Green-Blue (RGB) stego image. LSB, least significant bit

fulfil the condition and the embedding will be in the H plane. For pixel C, it belongs to the $(236^\circ, 237^\circ, 238^\circ)$ cluster of the *Blue* group and its one secret bit is 0; so the H_{steg} is the first hue in the cluster. Pixel D belongs to the $(254^\circ, 255^\circ, 256^\circ)$ cluster of the *Blue-Violet* group and its one secret bit is 1; so another bit will be taken, which is 0. Thus, the H_{steg} is the second hue in the cluster.

Next, all the planes are normalised back to the range $[0, 1]$ and the combination of them forms the HSV image that holds the secret. Lastly, the image is converted back to the RGB model in the range $[0, 1]$ and this stego image is sent to the receiver.

3.4 | Data extraction using the TPE method

For extracting the secret data from the stego image, the data extraction process of the proposed TPE method is performed

as explained in this section. First, the image is transformed to the HSV colour model. Then, the hue plane is normalised to the range $[0, 360]$ and both the saturation and the value planes are normalised to the range $[0, 255]$. The subsequent steps are described as follows:

For each pixel, we firstly check its saturation value. If its S is zero, we extract one secret bit from the LSB of the V plane. If not, we check whether its S and V are both above 20% or 51 in the range $[0, 255]$. If so, one secret bit is extracted from the LSB of the S plane; otherwise, one or two secret bits are extracted from the H plane using the extraction procedure of the HBE method, see Section 3.2. After scanning all the pixels in the image, the secret message is the concatenation of all the extracted bits.

We give a practical example of the TPE data extraction process by extracting the secret bits from the stego image in Figure 4. The image is first converted to the HSV space,

followed by normalising the three planes to the above-mentioned ranges. From pixel A to D, we notice that only the first pixel is unsaturated; so we extract the LSB of the V plane of pixel A as a secret bit. Pixel B is saturated and illuminated at values above 20%; so the LSB of the S plane is a secret bit. Pixel C is saturated at a value of less than 20% and the last pixel is illuminated at a value of less than 20%; therefore we extract secret bits from their hue planes. Since the cluster that the pixel C belongs to is $(236^\circ, 237^\circ, 238^\circ)$ and the H_{steg} is the first hue in the cluster, one secret bit is 0. Pixel D belongs to the $(254^\circ, 255^\circ, 256^\circ)$ cluster that has three members and the H_{steg} is the second hue in the cluster; so we extract two secret bits which are (10). Finally, all the bits are grouped to form the secret data.

4 | COMPARISONS AND ANALYSIS

In this section, we analyse the performance of the two proposed data hiding methods with respect to some of the existing state-of-the-art techniques that are different from reversible data hiding based on image interpolation [7]. The description of the platform and assessment metrics used is given in Section 4.1, followed by a discussion of the experimental results of the two techniques in Section 4.2.

4.1 | Experimental study

We used the MATLAB R2018a software running on 2.9 GHz Intel Core i7 CPU and 8 GB RAM as the platform for performing the experiments. We implemented the two proposed techniques as well as the following six existing methods: Karim et al. [12], simple LSB substitution in the S plane of the HSV model (LSB[S]) and simple LSB substitution in the V plane (LSB[V]), Muhammad et al. [18], HSV-MLSB [15] and Alwan et al. [20]. For the HSV-MLSB method [15], we chose the block size to be 2×2 as it provided the maximum embedding capacity for the method. The techniques were applied on 25 colour images of size 512×512 , which were

downloaded from the CVG-UGR image database [39] as shown in Figure 5. Additionally, the secret stream used was pseudo-random bits.

For evaluating the visual quality of a colour stego image S with respect to the corresponding original cover image C , we used the PSNR, in dB, as a metric. It requires the value of the mean square error (MSE) to be calculated independently for the three channels of the RGB model. The equations of the PSNR and the MSE are given in Equations (1) and (2), respectively, where the two images are of size $M \times N$ and the $C(i, j)$ and $S(i, j)$ are the values of the pixels at row i and column j in the two images. A high PSNR value of a stego image indicates that it is of high security quality.

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{(MSE_R + MSE_G + MSE_B)/3} \right) \quad (1)$$

$$MSE = \frac{\sum_{i=1}^M \sum_{j=1}^N (C(i, j) - S(i, j))^2}{M \times N} \quad (2)$$

The second performance metric used in the experiments is the embedding capacity in bits. It measures the maximum amount of the secret payload that the image can carry by means of a hiding technique. As it increased, the number of the embeddable bits increased.

TABLE 1 Performance evaluation of the proposed HBE and TPE methods in three colour spaces over 25 test images. The highest value for each metric is shown in boldface

Metric	HBE method			TPE method		
	HSV	HSI	HSL	HSV	HSI	HSL
Average capacity	367,502	367,350	367,462	310,201	339,388	329,756
Average PSNR	53.29	53.84	53.29	58.47	57.26	58.25

Abbreviations: HBE, hue-based embedding; HSI, Hue-Saturation-Intensity; HSL, Hue-Saturation-Lightness; HSV, Hue-Saturation-Value; PSNR, peak signal to noise ratio; TPE, three-planes embedding.

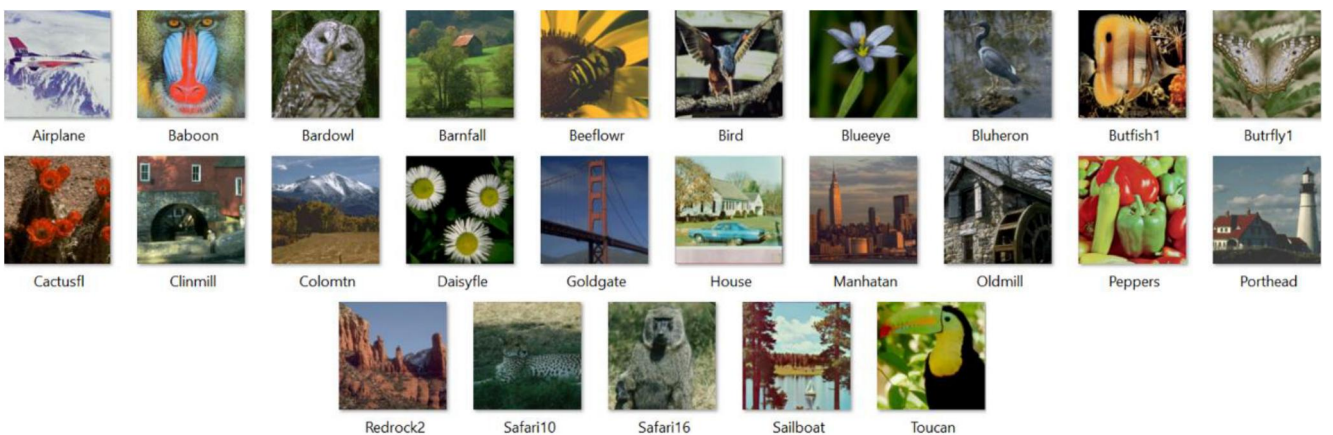


FIGURE 5 The 25 test images

TABLE 2 Comparison of PSNR values and embedding capacity values of the eight methods. The highest value for each metric for each image is shown in boldface

Test image	Metric	Karim et al. [12]	LSB (S)	LSB (V)	Muhammad et al. [18]	HSV-MLSB [15]	Alwan et al. [20]	Prop. HBE	Prop. TPE
Airplane	Capacity	262,144	262,144	262,144	262,144	131,072	262,144	392,246	364,249
	PSNR	55.93	55.22	51.27	50.77	54.28	50.02	56.16	61.27
Baboon	Capacity	=	262,119	=	=	=	=	389,128	295,241
	PSNR	55.90	56.88	51.71	50.02	54.72	49.97	48.43	57.24
House	Capacity	=	262,144	=	=	=	=	390,553	325,858
	PSNR	55.91	56.53	51.45	50.42	54.47	49.98	52.19	57.92
Peppers	Capacity	=	262,143	=	=	=	=	376,983	271,277
	PSNR	55.92	56	53.30	48.76	56.31	49.99	46.59	55.89
Sailboat	Capacity	=	262,144	=	=	=	=	385,485	298,724
	PSNR	55.91	57.82	52.13	49.89	55.16	49.96	51.94	58.77
Bardowl	Capacity	=	262,144	=	=	=	=	381,415	339,066
	PSNR	55.93	60.24	51.67	50.33	54.66	49.92	57.32	60.71
Beeflowr	Capacity	=	262,144	=	=	=	=	347,635	271,769
	PSNR	55.92	59.06	52.01	49.96	55	50.09	46.50	56.73
Blue-eye	Capacity	=	257,555	=	=	=	=	384,756	334,242
	PSNR	55.92	62.35	53.04	48.52	56.05	50.23	52.54	56.90
Cactusfl	Capacity	=	246,681	=	=	=	=	366,414	295,237
	PSNR	55.92	59.21	52.68	48.87	55.72	50.05	49.37	56.78
Barnfall	Capacity	=	262,144	=	=	=	=	393,064	279,999
	PSNR	55.91	62.01	51.86	49.89	54.89	49.74	51.97	61.66
Goldgate	Capacity	=	260,987	=	=	=	=	384,712	283,878
	PSNR	55.91	60.04	51.85	49.56	54.88	49.79	51.07	59.33
Porthhead	Capacity	=	262,144	=	=	=	=	325,750	327,145
	PSNR	55.90	59.21	51.21	50.98	54.23	49.75	58.89	57.69
Bird	Capacity	=	262,144	=	=	=	=	383,323	348,715
	PSNR	55.92	59.82	51.52	50.26	54.54	50.02	57.82	60.91
Safari16	Capacity	=	262,144	=	=	=	=	393,064	328,737
	PSNR	55.92	60.40	51.16	50.58	54.18	49.79	57.65	61.76
Butrffy1	Capacity	=	262,144	=	=	=	=	393,064	299,279
	PSNR	55.92	59.81	51.42	50.28	54.42	49.96	54.33	60.73
Manhatan	Capacity	=	262,144	=	=	=	=	368,659	322,495
	PSNR	55.93	59.18	51.45	50.38	54.46	50.01	54.39	58.25
Daisyfle	Capacity	=	181,480	=	=	=	=	265,723	320,991
	PSNR	55.91	61.23	52.09	49.16	55.11	50.47	54.26	55.09
Butfish1	Capacity	=	206,116	=	=	=	=	301,646	278,711
	PSNR	55.92	58.83	52.09	50.01	55.11	49.74	48.35	55.13
Toucan	Capacity	=	208,071	=	=	=	=	308,608	273,272
	PSNR	55.91	59.22	52.34	49.31	55.36	49.93	46.71	55.35
Oldmill	Capacity	=	236,544	=	=	=	=	335,228	354,020

(Continues)

TABLE 2 (Continued)

Test image	Metric	Karim et al. [12]	LSB (S)	LSB (V)	Muhammad et al. [18]	HSV-MLSB [15]	Alwan et al. [20]	Prop. HBE	Prop. TPE
Redrock2	PSNR	55.91	60.96	51.39	50.28	54.38	50.27	60.56	58.22
	Capacity	=	262,144	=	=	=	=	391,189	276,774
Bluheron	PSNR	55.91	57.79	51.56	50.15	54.58	50.05	51.34	57.52
	Capacity	=	262,144	=	=	=	=	361,779	349,409
Colomtn	PSNR	55.91	62.26	51.16	50.69	54.15	49.63	60.60	59.77
	Capacity	=	262,144	=	=	=	=	386,948	286,509
Clinmill	PSNR	55.92	58.22	51.67	50.21	54.67	50.08	51.97	58.32
	Capacity	=	259,501	=	=	=	=	387,136	313,861
Safari10	PSNR	55.91	60.14	51.33	50.18	54.35	49.71	54.64	59.50
	Capacity	=	262,144	=	=	=	=	393,064	315,590
Average	PSNR	55.91	62.33	51.50	50.22	54.52	49.94	56.55	60.36
	Capacity	262,144	252,534	262,144	262,144	131,072	262,144	367,502	310,201
	PSNR	55.92	59.39	51.79	49.99	54.81	49.96	53.29	58.47

Abbreviations: HBE, hue-based embedding; HSV, Hue-Saturation-Value; LSB, least significant bit; PSNR, peak signal to noise ratio; TPE, three-planes embedding.

4.2 | Analysing the data hiding results

In this section, we examine the outcomes of the two proposed techniques and the six methods, which were involved in the experiments, in terms of the visual quality and the embedding capacity.

First, we embedded the secret message into all the test images using our proposed HBE and TPE methods in three hue-based colour spaces: HSV, HSI and HSL. Using the TPE method, the V plane in HSV was considered as the I and L planes in the other two spaces. According to Table 1, the HSV model achieved the highest average capacity in the HBE method, while it scored the highest average PSNR in the TPE method. On the contrary, the HSI model reached the utmost quality in HBE, while it attained the maximum capacity in TPE. HSL was approximately in the middle in all cases.

The reason the HSI model provided the highest PSNR in the HBE method is that it obtained the lowest payload in the method and there is usually an inverse relationship between the amount of the payload and the visual quality. Likewise, the HSV model provided the highest PSNR in the TPE method because it attained the lowest embedding capacity in the method.

On comparing the two methods in all the spaces, the HSV model achieved the best average capacity and PSNR. This is one of the reasons that we chose it as the colour model of our proposed methods.

The second experiment was the implementation of all the eight competed techniques. The performance results are given in Table 2. From the table, it can be observed that our HBE could embed at least 265,723 bits and maximum 393,064 bits, while TPE could hide at least 271,769 bits reaching at most 364,249. It is obvious that both of our methods achieved the best embedding capacity among all the other methods in all the

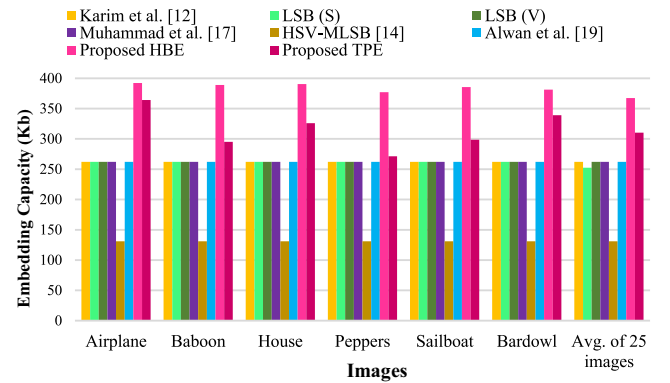


FIGURE 6 Comparison of embedding capacity values of the eight methods. HBE, hue-based embedding; HSV, Hue-Saturation-Value; LSB, least significant bit; TPE, three-planes embedding

images. The reason is that all the other techniques are 1LSB-based; so they embed maximum one bit into a pixel, whereas our methods might embed one or two bits into a pixel. In most of the images, our HBE method was better than our TPE method in terms of the embedding capacity. But the latter was better than the former in terms of the visual quality of the images. Figure 6 illustrates the embedding capacity values of all of the eight techniques.

By looking at our HBE method, it can be observed that it attained PSNR values ranging from 46.50 to 60.60 dB, which are considered high values of the image quality. Compared with the existing methods, the average security quality of HBE is better than LSB(V) and the methods proposed by Muhammad et al. [18] and Alwan et al. [20]. In Figure 7, we provide a diagrammatic representation of the visual quality of the eight methods.

Our TPE method achieved PSNR values of minimum 55.09 dB and maximum 61.76 dB; so the visual quality of

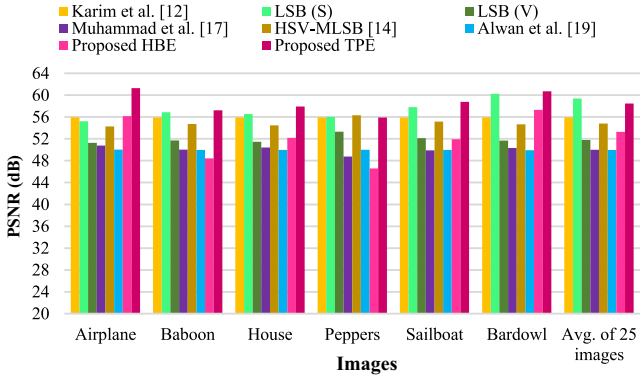


FIGURE 7 Comparison of peak signal to noise ratio values of the eight methods. HSV, Hue-Saturation-Value; LSB, least significant bit

the images is very high. Additionally, the average quality of the method was the best among the five existing techniques by Karim et al. [12], LSB(V), Muhammad et al. [18], HSV-MLSB [15] (with block size = 2×2) and Alwan et al. [20].

On the other hand, the average PSNR of the LSB(S) method was higher than our TPE method but by less than 1 dB, and ours was better than the former in nine images including four standard images: airplane, baboon, house and sailboat. The average PSNR of the LSB(S) method was higher than the two approaches proposed for most of the images because it embedded less payload and the maximum difference between the value of a pixel before and after the embedding was 1, while in our methods the difference could reach 2, which might reduce the quality.

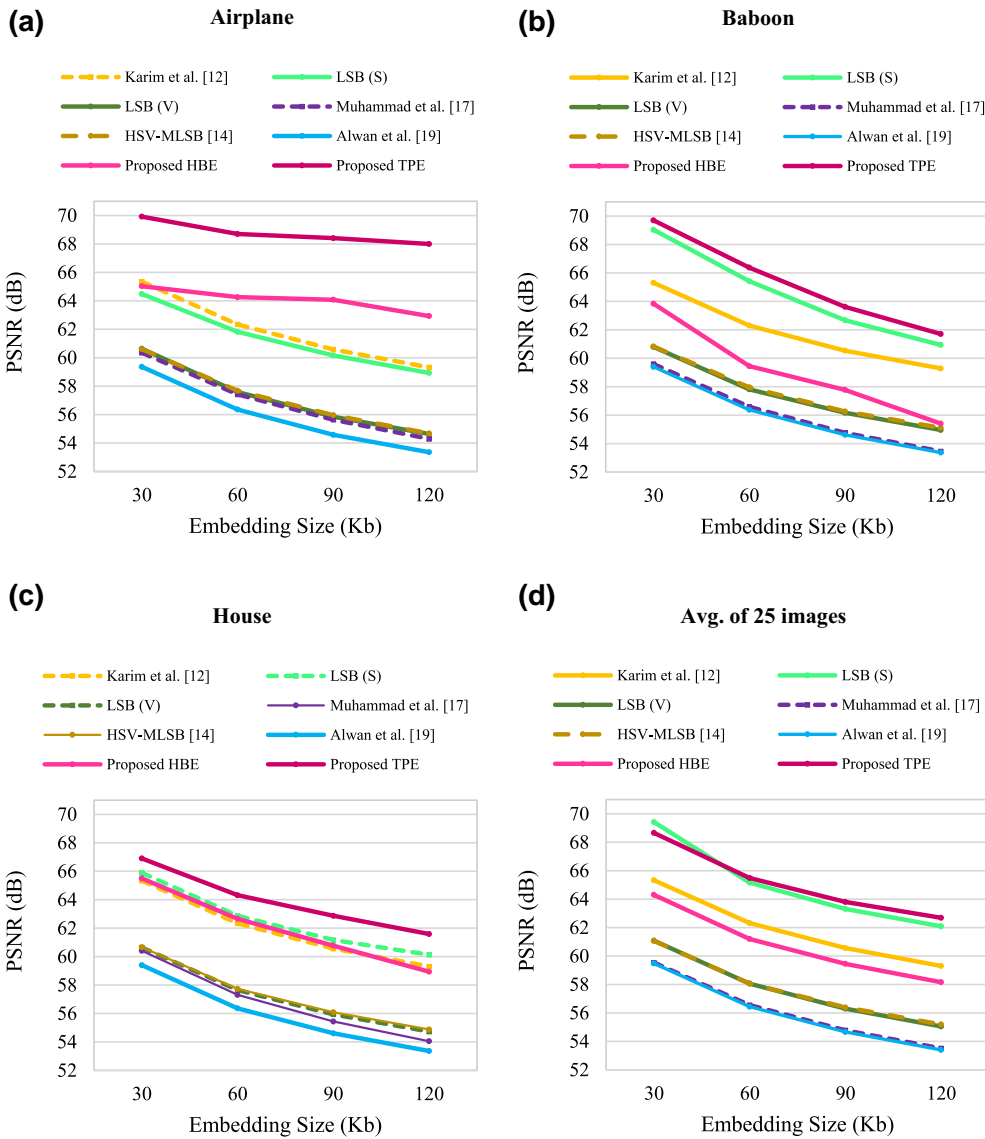


FIGURE 8 Comparison of peak signal to noise ratio (PSNR) values of the eight methods for varying embedding size in test images (a) Airplane, (b) Baboon, (c) House, (d) Average of 25 images. HSV, Hue-Saturation-Value; LSB, least significant bit; TPE, three-planes embedding

The high visual quality of the LSB(S) method and the high payload of our HBE method made us compromise both in our TPE method by using a percentage of 20% to be the threshold of embedding in either the H or S planes. Increasing this threshold will increase the embedding capacity but reduce the visual quality and vice versa.

The PSNR results of the techniques used by varying the size of the embedded bits are shown in Figure 8. The payload sizes tested were chosen by taking into consideration that the HSV-MLSB [15] method in our experiments had a maximum capacity of 131,072 bits. Therefore, the selected sizes did not exceed this capacity. As we can see in the figure, our proposed TPE method scored the highest average PSNR among all the methods in embedding 60, 90 and 120 kilo bits of data. The proposed TPE method provided a smaller average PSNR than the LSB(S) approach when the embedding size was 30 Kb because an outlier PSNR value of 89.16 dB appeared when embedding this payload in the 'Blue-eye' image using the LSB (S) method. This outlier led to increasing the methods average quality more than ours.

5 | CONCLUSION

In this article, we have proposed two enhanced data hiding techniques that use the HSV colour model to improve the embedding and security performance. We used the HSV space because it is more suitable for our methods than the HSI and HSL models, as verified in the experiments. Both of the techniques proposed outperformed the embedding capability of the methods in [12, 15, 18, 20] as well as the LSB(S) and LSB(V) methods with high image quality. These existing competing methods are either in RGB, HSI, HSV or YCbCr colour spaces. In our HBE method, we utilised the hue plane for embedding one or two bits in all the saturated pixels; so the hiding capacity of the method could reach 393 Kb and the security quality could reach 60 dB. Our TPE method was a combination of the HBE, LSB(S) and LSB(V) techniques in which we applied LSB(V) on the unsaturated pixels, LSB(S) on the pixels that are of both high saturation and high brightness and HBE on the rest of the pixels. This combination made the TPE method have a visual quality of at least 55 dB in all the test images with a payload of at least 271 Kb. The experiments have proven that on average, our HBE method could embed more bits than our TPE method and its average PSNR is higher than LSB(V) and the methods in [18, 20]. Additionally, our TPE technique achieved the best average visual quality compared with HBE, LSB(V) and the methods in [12, 15, 18, 20].

In the future, we plan to further increase the visual quality of the TPE technique by replacing the simple LSB substitution method in the saturation and value planes with more improved methods.

ACKNOWLEDGMENTS

Thanks to Umm Al-Qura University (UQU) for supporting this work. There is no funding source.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

ORCID

Adnan Gutub  <https://orcid.org/0000-0003-0923-202X>

REFERENCES

- Al-Juaid, N., Gutub, A.: Combining RSA and audio steganography on personal computers for enhancing security. *SN Appl. Sci.* 1, 830 (2019). <https://doi.org/10.1007/s42452-019-0875-8>
- Gutub, A., Al-Ghamdi, M.: Hiding shares by multimedia image steganography for optimized counting-based secret sharing. *Multimed. Tool. Appl.* 79(11–12), 7951–7985 (2020). <https://doi.org/10.1007/s11042-019-08427-x>
- Al-Juaid, N., Gutub, A., Khan, E.: Enhancing PC data security via combining RSA cryptography and video based steganography. *J. Inf. Secur. Cybercrimes Res.* 1, 5–13 (2018). <https://doi.org/10.26735/16587790.2018.006>
- Alanazi, N., Khan, E., Gutub, A.: Functionality-improved Arabic text steganography based on unicode features. *Arabian J. Sci. Eng.* 45(12), 11037–11050 (2020). <https://doi.org/10.1007/s13369-020-04917-5>
- Alanazi, N., Khan, E., Gutub, A.: Efficient security and capacity techniques for Arabic text steganography via engaging unicode standard encoding. *Multimed. Tool. Appl.* 80, 1403–1431 (2020). <https://doi.org/10.1007/s11042-020-09667-y>
- Shi, Y., et al.: Reversible data hiding: advances in the past two decades. *IEEE Access.* 4, 3210–3237 (2016)
- Hassan, F., Gutub, A.: Efficient reversible data hiding multimedia technique based on smart image interpolation. *Multimed. Tool. Appl.* 79(39–40), 30087–30109 (2020). <https://doi.org/10.1007/s11042-020-09513-1>
- Hassan, F., Gutub, A.: Novel embedding secrecy within images utilizing an improved interpolation-based reversible data hiding scheme. *J. King Saud Univ. Comput. Inf. Sci.* (2020, in press). <https://doi.org/10.1016/j.jksuci.2020.07.008>
- Li, Q., et al.: A novel grayscale image steganography scheme based on chaos encryption and generative adversarial networks. *IEEE Access.* 8, 168166–168176 (2020)
- Wang, Y., et al.: Non-additive cost functions for color image steganography based on inter-channel correlations and differences. *IEEE Trans. Inf. Forensics Secur.* 15, 2081–2095 (2019)
- Liao, X., et al.: A new payload partition strategy in color image steganography. *IEEE Trans. Circ. Syst. Video Technol.* 30(3), 685–696 (2020)
- Masud Karim, S., Rahman, M., Hossain, M.: A new approach for LSB based image steganography using secret key. In: *Proceedings of the International Conference on Computer and Information Technology (ICCIT)*, pp. 286–291. Dhaka (2011)
- Gutub, A.: Pixel indicator technique for RGB image steganography. *J. Emerg. Technol. Web Intell.* 2(1), 56–64 (2010). <https://doi.org/10.4304/jetwi.2.1.56-64>
- Muhammad, K., et al.: CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method. *Multimed. Tool. Appl.* 76, 8597–8626 (2016)
- Muhammad, K., et al.: Evaluating the suitability of color spaces for image steganography and its application in wireless capsule endoscopy. In: *Proceedings of the International Conference on Platform Technology and Service (PlatCon)*, pp. 1–3. Jeju (2016)
- Muhammad, K., et al.: Image steganography using uncorrelated color space and its application for security of visual contents in online social networks. *Future Generat. Comput. Syst.* 86, 951–960 (2018)
- Hussein, Q., Saadi, A., Nada, Q.: The efficiency of color models layers at color images as cover in text hiding in text hiding. *Tikrit J. Pure Sci.* 21(1), 130–139 (2016)
- Muhammad, K., et al.: A novel image steganographic approach for hiding text in color images using HSI color model. *Middle East J. Sci. Res.* 22, 647–654 (2014)

19. Muhammad, K., et al.: Image steganography for authenticity of visual contents in social networks. *Multimed. Tool. Appl.* 76(18), 18985–19004 (2017)
20. Alwan, Z., Farhan, H., Mahdi, S.: Color image steganography in YCbCr space. *Int. J. Electr. Comput. Eng.* 10(1), 202–209 (2020)
21. Evsutin, O., et al.: An adaptive algorithm for the steganographic embedding information into the discrete fourier transform phase spectrum. *Adv. Intell. Syst. Comput.* 451, 47–56 (2016)
22. Ibraheem, N., et al.: Understanding color models: a review. *ARPN J. Sci. Technol.* 2(3), 265–275 (2012)
23. Saravanan, G., Yamuna, G., Nandhini, S.: Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. In: *Proceedings of the 2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 462–466. Melmaruvathur (2016)
24. Tsai, C., et al.: Lossless data hiding of color images using pixel decomposition and phase difference. *J. Inf. Sci. Eng.* 23(5), 1481–1498 (2007)
25. Muhammad, K., et al.: A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image. *Multimed. Tool. Appl.* 75(22), 14867–14893 (2016)
26. Kumar, S., et al.: A novel approach to hide text data in colour image. In: *Proceedings of the International Conference on Reliability, IEEE Infocom Technologies and Optimization: Trends and Future Directions (ICRITO)*, pp. 577–581. Noida (2018)
27. Swain, G., Lenka, S.: A better RGB channel based image steganography technique. *Commun. Comput. Inf. Sci.* 270(PART II), 470–478 (2012)
28. Das, P., Kar, N.: ILSB: indicator-based LSB steganography. In: Jain, L., Patnaik, S., Ichalkaranje, N. (eds.) *Intelligent Computing, Communication and Devices*, vol. 309, pp. 489–495. Springer, New Delhi (2015)
29. Pandey, J., et al.: Pixel indicator steganography technique with enhanced capacity for RGB images. In: *Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 738–743. Madurai (2019)
30. Ioannidou, A., Halkidis, S., Stephanides, G.: A novel technique for image steganography based on a high payload method and edge detection. *Expert Syst. Appl.* 39(14), 11517–11524 (2012)
31. Zhang, X., Wang, S., Zhou, Z.: Multibit assignment steganography in palette images. *IEEE Signal Process. Lett.* 15, 553–556 (2008)
32. Grover, N., Mohapatra, A.: Digital image authentication model based on edge adaptive steganography. In: *Proceedings of the 2013 2nd International Conference on Advanced Computing, Networking and Security (ADCONS)*, pp. 238–242. Mangalore (2013)
33. Rashid, R., Majeed, T.: Edge based image steganography: problems and solution. In: *Proceedings of the 2019 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pp. 1–5. Sharjah (2019)
34. Qin, X., et al.: A novel steganography for spatial color images based on pixel vector cost. *IEEE Access.* 7, 8834–8846 (2019)
35. Wang, Y., et al.: Non-additive cost functions for JPEG steganography based on block boundary maintenance. *IEEE Trans. Inf. Forensics Secur.* 16, 1117–1130 (2021)
36. Rabie, T., Kamel, I.: High-capacity steganography: a global-adaptive-region discrete cosine transform approach. *Multimed. Tool. Appl.* 76(5), 6473–6493 (2017)
37. Ahani, S., Ghaemmaghami, S.: Colour image steganography method based on sparse representation. *IET Image Process.* 9(6), 496–505 (2015)
38. Smith, A.: Color gamut transform pairs. *Comput. Graph. (ACM)*. 12, 12–19 (1978)
39. CVG-UGR Image Database. <http://decsai.ugr.es/cvg/dbimagenes/> (2020). Accessed December 2020

How to cite this article: Hassan, F.S., Gutub, A.: Improving data hiding within colour images using hue component of HSV colour space. *CAAI Trans. Intell. Technol.* 1–13 (2021). <https://doi.org/10.1049/cit2.12053>