


A Novel LSB Steganography Technique Using Image Segmentation


Yasir Yakup Demircan

(Üsküdar University, Istanbul, Turkey)

 <https://orcid.org/0000-0002-4641-3930>, yasirdemircan@gmail.com

Serhat Ozekes

(Marmara University, Istanbul, Turkey)

 <https://orcid.org/0000-0002-7432-0272>, serhat.ozekes@marmara.edu.tr

Abstract: Steganography is a process to hide data inside a cover file mostly used in media files like image, video, and audio files. Least significant bit (LSB) steganography is a technique where the least significant bits of pixels are used for information hiding. The purpose of using only those bits is to minimize the visual impact of the hidden data on the image file. LSB technique of steganography is one of the most popular forms of steganography available today. As a result, various steganalysis techniques are developed for this steganography technique. One of them is the visual analysis of pixels through pixel modifications to expose hidden data in a visual manner. The proposed method achieves resistance to this attack using an image segmentation model and extracting the most texture-complex areas of an image and hiding information in these specific areas as pseudo-randomized least significant bit replacements. As the outcome of the study, an alternative approach to LSB steganography that results competitively with existing methods is provided.

Keywords: Steganography, Information hiding, Visual attack, LSB embedding, Steganalysis

Categories: C.2.0

DOI: 10.3897/jucs.105702

1 Introduction

Steganography is the art and science of concealing information into ordinary-looking cover files which are often digital media files. This process aims to protect covert communication from third parties. The main strength of steganography is the way it conceals the secret data's existence thus aiming to eliminate a privacy breach [Cheddad et al., 2010]. On the other hand, cryptography has secret data in plain sight open to attack and its aim is to reinforce the data against attacks from third parties. Those mentioned data-protecting techniques are often used together to fix each other's weak points [Mishra and Bhanodiya, 2015]. This kind of combined approach can even be seen in low-power budget applications like Internet of Things (IoT) security [Khari et al., 2020].

Steganography as a word comes from the combination of two Greek terms “stegos” and “graphia” which means cover and writing respectively [Hariri et al., 2011]. The usage of steganography dates back to 440 B.C., and ancient methods at that time include wax tablets with hidden messages and hidden text beneath stamps. As relatively recent

examples both World Wars and Vietnam War are the conflicts in which steganography has been used.

The Least Significant Bit (LSB) steganography method, when used alone is susceptible to various steganalysis techniques like statistical steganalysis methods and visual filtering attacks. In this study, we hypothesize a method that makes the least significant bit of steganography resistant against the visual attack method found on common steganalysis applications [Westfeld and Pfitzmann, 2000] without introducing additional compromises on detectability with statistical steganalysis and image distortion tests. This study aims to obtain an alternative steganography method that matches the existing methods in terms of statistical and distortion test results, the proposed method also utilizes neural network image segmentation as a way to resist visual attack.

The proposed method contributes to the field as it shows a new way to reinforce the existing LSB steganography method using an artificial intelligence image processing technique. We believe that our method is an advancement in the field of steganography and can open up new possibilities for secure and covert communication.

By using this kind of segmentation in the algorithm we introduce additional variables to obtain in order to extract the data, which are the training data of the segmentation model and the structure of the used segmentation model. In order to obtain the data both of these variables should be known by the sender and the receiver side which makes the proposed method more resistant to a brute forcing approach than methods that only use a password to protect the embedded data.

This paper is further organized into the following sections: Section 2 explains various steganography methods and the LSB method used for the proposed technique is explained with an example. Section 3 gives the information about the past studies in the field. Section 4 explains the details of the proposed method, in addition to that graphs for the framework is given. Section 5 explains the testing methods. Section 6 provides tests results and finally, the conclusion and the further work is discussed in Section 7.

1.1 A brief introduction to image segmentation

Image segmentation is a growing area of interest in image processing and computer vision. It provides a vital framework for picture identification. An input image is categorized based on several comparable criteria in order to extract the area that people are interested in. In addition, it provides the groundwork for understanding image analysis as well as picture feature extraction and recognition [Yuheng and Hao, 2017]. There are various image segmentation methods popularly used which can be categorized as region-based segmentation, edge detection segmentation, clustering-based segmentation, and convolutional neural network (CNN) based segmentation [Kaur and Kaur, 2014].

Image segmentation aims to transform an image into a simpler or different form that can be more readily interpreted and examined. The common application of image segmentation is to identify the location of objects and their boundaries in images. Segmenting the foreground pixels from the background pixels is an example of this process. In essence, image segmentation involves assigning a category to each pixel in an image based on the similarity of their attributes, such as color or texture.

Image segmentation is useful for various domains, such as medical imaging, autonomous vehicles, and satellite imagery, among others. For instance, in medical imaging, image segmentation can help to detect tumors and other abnormalities or identify diseases.

In this study, we are utilizing a pre-trained DeepLabV3 convolutional neural network method of image segmentation to extract the area used for the proposed extracting and embedding algorithms. The reason for using the neural network-based approach in this study is that a neural network can be trained with different variations of data and this variation will influence the segmentation process. Because of this reason different models and training variations can be used to secure the segmentation information if the model is trained to segment specific parts of the image. Thus providing additional security to the steganography process. The image segmentation model used for the results is discussed in section 6 of this paper.

Figure 1 shows an example image and the segmentation map obtained from the image.



Figure 1: Example image (Left) and the segmentation map (Right) of the image

1.2 A brief introduction to steganalysis

Steganalysis is the science of uncovering and analyzing a stego image. [Fridrich et al., 2001] The aim of steganalysis is that break the secrecy of steganography by revealing enough information about embedding in the carrier medium. Steganalysis is used in some areas like computer forensics and law enforcement, it is useful for acquiring evidence about a criminal activity involving steganography. Another usage of steganalysis is to assess and find flaws in steganographic systems and help them improve upon their weaknesses [Nissar and Mir, 2010]. Steganalysis studies began in the late 1990s. The research by Johnson and Jajodia [Johnson and Jajodia, 1998] is the first reported work in steganalysis. Different techniques of steganalysis will be discussed in further sections of this study.

2 Image Steganography Methods

Image steganography can be categorized under two major categories which are the transform domain (DWT, DCT...) and spatial domain (LSB, PVD...)[Alyousuf et al., 2020]. Spatial domain steganography directly modifies pixels to embed the desired message into the carrier image. These modifications can use in various ways from

logarithmic transforms to directly modifying bits in the pixel [Sharma and Kumar, 2015]. One of the most used techniques in this domain is LSB embedding. Transform domain techniques on the other hand utilizes the orthogonal transform of the image for data embedding modifications instead of direct pixel values [Sharma and Kumar, 2015].

2.1 LSB Steganography Method

The Least Significant Bit (LSB) steganography method is a spatial domain steganography method, in which the proposed method is based on modifying the rightmost bit of a color channel in a pixel's binary representation to embed a message into the selected cover image.

To hide a decimal number 10 as binary "1010" in the LSB method, two pixels of an RGB image are needed and the modifications will be done as following :

Example 6 bytes of an RGB image :

(00100110, 01011001, 10110000)

(00101001, 10101111, 00110101)

Modified example bits (shown as bold) with hidden value :

(0010011**1**, 0101100**0**, 1011000**1**)

(0010100**0**, 10101111, 00110101)

In the example indicated 4 bits of the example 6 bytes are modified to contain the desired binary sequence. This change to the selected pixels is close to the original and thus does not make a significant enough difference to be noticed by the eye [Arya and Soni, 2018].

3 Previous Work

There are many studies in the field concerning data hiding techniques. The most known one is the LSB steganography technique in images [Ansari et al., 2019]. Segmenting images with various techniques to find the most suitable places in the image for data hiding is proposed in numerous studies. Previous works explained below are a collection of LSB, image segmentation, and neural network steganography methods:

Yang et al. used histogram shifting to hide data inside a textured area and use the contrast enhancement technique on the image to increase the perceived quality of the image. Medical images are used to test this steganography method [Yang et al., 2015]. Duan et al. trained deep neural networks which include a hiding network and an extraction network. Sender uses the hiding network to embed an image into another cover image and the receiver uses the extraction network to retrieve the secret image from the cover image. This neural network uses the U-Net structure and aims to create a high-capacity steganography method [Duan et al., 2019]. Nosrati et al. propose a heuristic genetic algorithm for image steganography. In this study genetic algorithm is used to select the most suitable places in the image for data hiding that requires a minimum change to keep cover image characteristics intact [Nosrati et al., 2015]. Abuzanounch and Hadwan proposed a steganography technique using feature selection to find characteristics of the cover image and hide data randomly inside the cover image. The method produces a complex key to rearrange used bits for hiding. This

method aims to complicate the steganalysis process and deter attackers from compromising the hidden data [Abuzanouneh and Hadwan, 2021]. Al-Ahmad et al. developed a technique called Modified Deep Hiding Extraction Algorithm (MDHEA) which uses color image segmentation to select the most appropriate places to hide data and Blue Smoothing Algorithm (BSA) to hide the noise generated by the data embedding [Al-Ahmad et al., 2021]. Bawaneh and Obeidat proposed a grayscale image steganography using Least Significant Bit(LSB) embedding and image segmentation. The method is secured by a master key which generates an area selection key, pixel selection key, and cryptography key. The proposed method met the main requirements for a steganography technique like security, modification rate, and capacity [Bawaneh and Obeidat, 2016]. Luo et al. created a coverless image steganography technique utilizing Faster RCNN to detect objects inside the image dataset and create connections between objects and binary sequences then for sequence generation a novel mapping rule is proposed which is based upon filtered robust object labels. This way an image can generate a binary sequence with the use of object recognition. The transmitted image is unmodified in this method as a result can resist steganalysis tools [Luo et al., 2020]. Zaini proposed a method to secure LSB2 image steganography utilizing wavelet packet decomposition. In the resulting decomposition, one of the segments will be used to cover the data. The method can be used for long or short messages and tested using Mean Square Error (MSE), Peak Signal to Noise Ratio(PSNR) image metrics as well as algorithm performance [Zaini, 2021]. Hempstalk proposed two novel steganography methods called FilterFirst and BattleSteg both algorithms are based on LSB embedding. The first algorithm called FilterFirst uses various edge detection algorithms to find the best place to hide data, even though it is effectively resistant to steganalysis attacks the algorithm needs additional security measures for it to be considered secure. The BattleSteg algorithm combines Filterfirst and Hideseek algorithms to create a secure and harder to detect algorithm, it utilizes a BattleShip game-inspired pattern for hiding information securely. Both methods are tested against LSB embedding with no modification and the HideSeek algorithm [Hempstalk, 2006].

4 Design of the Proposed Algorithm

The proposed algorithm in this paper consists of two parts, the embedding algorithm, and the extracting algorithm. Proposed algorithms are built using a pipeline of two separate parts, one of which is the segmentation model algorithm written in Python and used by both algorithms without modification and it creates the segmentation map that is required for both embedding and extracting algorithms. The other ones are embedding and extracting algorithms respectively which are written in Javascript in a form of separate web applications.

4.1 Embedding Phase

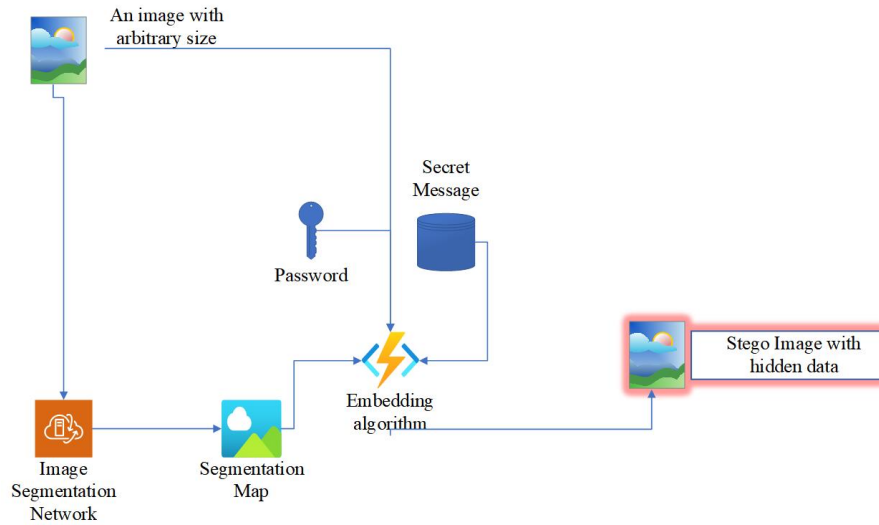


Figure 2: Framework of the embedding algorithm

The embedding algorithm starts with the step of getting the segmentation map from the original image. Any segmentation neural network can be used in this phase, but the training data and the model itself used for embedding the data should be exactly the same for a successful extraction process along with the provided password. The next step after the segmentation is creating an RGB array of the provided image, which consists of red, green, and blue values for each pixel in the image.

After we obtain RGB values, a boolean array is created using the segmentation provided by the neural network. This array consists of true and false values associated with each pixel in the segmentation map as foreground(included) and background(excluded) respectively. This way the most noiseless side of the image is excluded by the embedding algorithm which is the first part of resisting the visual attack.

After the boolean array is created we process the provided password using the SHA-256 hashing algorithm [Rachmawati et al., 2018]. After this step the obtained hash value is processed again with the same hashing algorithm five times in total each new hashing process is concatenated with the original result to create a longer randomized hash value the final longer hash value is repeated according to the hidden message bit size. The step after getting the aforementioned final hash is creating a binary value array consisting of three bit values for each hexadecimal hash digit. Three bit values are obtained by getting the remainder of each hash digit by division to eight then the result will be stored as a three bit binary value in the said array.

The last array we are using will be obtained from the three bit value array which consists of the most repeated bit in three bits and it will be used to XOR hidden bits and randomize the exclusion of a color channel to avoid using the same channel excessively which would result in easy visibility in the visual attack.

4.2 Extraction Phase

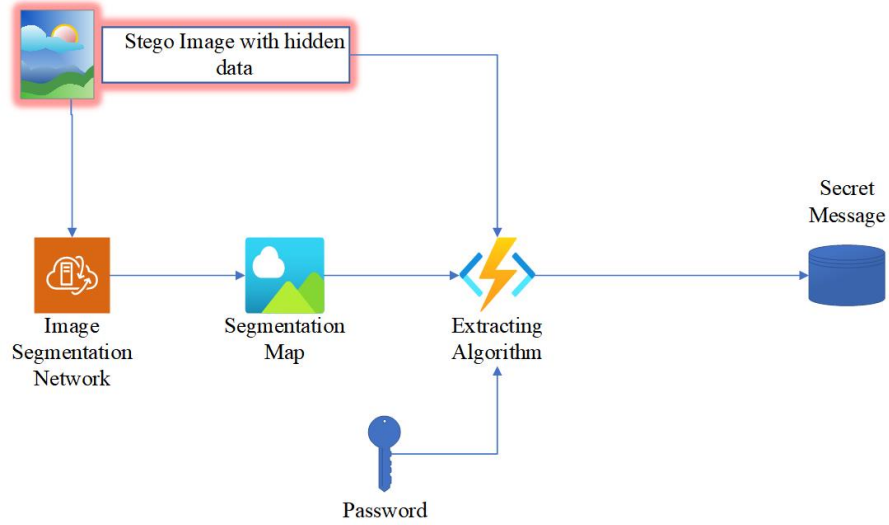


Figure 3: Framework of the extraction algorithm

The extracting algorithm follows the same steps as the embedding algorithm as long as the segmentation method and the password is exactly the same as the embedding algorithm, the embedded data can be extracted from the stego image successfully.

The proposed algorithm achieves 1 bpp (bits per pixel) hiding capacity with half of the image segmented as background. Which can increase or decrease depending on the segmented amount and the trained neural network. Exactly the same trained segmentation network is needed to reproduce pixel-accurate segmentation maps for both embedding and extracting algorithms. As a result that both the model and training data for the segmentation network and the password used in the embedding process should be known for extracting the data successfully.

5 Testing Methods

The testing methods we used in this paper can be classified into three major categories, distortion measurements, which are used to discover the existence of a steganography embedding using the distortion of the stego image relative to the original image, statistical steganalysis methods, which are used to discover the length and/or existence of the stego-message inside an image. Lastly, we use the visual attack method which adjusts pixel values in a way to show the LSB embedding in a visual manner.

All the aforementioned methods are used to obtain information about a stego-message inside an image. The tests used to evaluate steganography methods in this study are explained below. These tests will give us an objective way to evaluate the proposed method and the other methods used to compare it.

5.1 Distortion measurements

In this category of metrics, the stego image and original cover image are compared for a score and this score reflects the change of distortion between the two images. To successfully use these metrics for steganalysis the original image source should be known and reachable. The following distortion metrics are used to test the proposed method against other methods.

5.1.1 Mean squared error (MSE)

MSE is calculated using the following formula [Tiwari and Shandilya, 2010] :

$$MSE = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (p_{ij} - q_{ij})^2$$

The p_{ij} and q_{ij} values are the pixel values of original and stego images respectively, at the i^{th} row and j^{th} column.

MSE value is desired to be low as possible in steganography methods. When the MSE value is zero this means the original image and the stego image are identical.

5.1.2 Peak signal-to-noise ratio (PSNR)

The PSNR is calculated using the following formula [Tiwari and Shandilya, 2010] :

$$PSNR = 10 \times \log_{10} \frac{M^2}{MSE}$$

PSNR is used to measure the distortion of the stego image in decibels (dB) the higher the value is the lesser distortion in the image. The value M is the maximum pixel value in the image, which is 255 for the color images using 3 bytes of data to represent pixels [Pradhan et al., 2016].

5.1.3 Structural Similarity Index (SSIM)

The SSIM metric provides an accurate measurement of perceived distortion between the original and stego images. To assess the similarity of two images, the metric examines how the brightness and contrast of small regions of pixels vary within each image and then compares them. It is based on the idea that the human visual system is very good at detecting structural differences in a scene [Wang et al., 2004].

5.2 Statistical Steganalysis Methods

Keeping the information of secret data's existence is the primary objective of a steganography method [Das and Tuithung, 2012]. Statistical steganalysis methods aim to expose the existence of a secret message inside the steganography applied medium. The aforementioned tests do not require the original image before steganalysis to be obtained for applying them. This aspect of the statistical steganalysis tests makes them

a great tool to use on suspected images. Even though they do not need the original image to function, keeping original images as a baseline will help to eliminate false positive results.

5.2.1 RS Analysis

RS analysis is a statistical steganalysis first proposed by Fridrich, et al. [Fridrich, Goljan and Du, 2001]. The test is used to estimate the hidden data length in randomly scattered LSB-embedded stego images. In this method, the LSB of the stego image is altered and compared with the non-altered stego image. The comparison is done by classifying pixels of both images as regular and singular, the count difference of regular and singular pixels in altered least significant bytes and non-altered least significant bytes of images increases as the hidden data length increases [Boehm, 2014].

5.2.2 Sample Pairs Analysis

Sample Pairs Analysis proposed by [Dumitrescu et al., 2003] is another type of steganalysis to estimate hidden data length in stego images. The technique works by comparing groups of sample pairs, known as trace multisets, that should have the same count of pairs if the signal is original. Random LSB embedding alters the count of pairs in different trace multisets with some chances, and this changes the statistical relations among them. The technique utilizes finite state machines those model previously mentioned statistical relations.[Hempstalk, 2006].

5.2.3 Primary Sets Analysis

Primary sets analysis is another steganalysis method that aims to find the hidden data length of the randomly scattered embedded data for LSB based steganography methods. This method analyses the change of cardinality by LSB embedding within some subset of pixels inside the image. The key concept of the analysis is explained below.

Subsets X , Y , and P are determined by following rules :

- P is the set of all pixel pairs.
- X is the set of pairs determined by the rule, $(u, v) \in P$ such that v is odd and $u > v$ or v is even and $u < v$
- Y is the set of pairs determined by the rule, $(u, v) \in P$ such that v is odd and $u < v$ or v is even and $u > v$

The length of the LSB embedded data can be detected by the assumptions about the natural images and randomly scattered pixels using the sets determined above [Dumitrescu et al., 2002].

5.2.4 Chi-square Attack

The chi-square attack was used by [Westfeld and Pfitzmann, 2000] as a steganalysis tool to detect LSB embedding inside images. The Chi-square method is more successful at detecting sequential embedded LSB steganography rather than randomly embedded LSB steganography [Karampidis et al., 2018]. This type of steganalysis works by comparing the stego image's frequency distribution with the expected frequency

distribution of the image using Chi-square statistical analysis [Hogg, 1957]. If the two distributions are significantly different, then the steganalyst can conclude that the suspect image is likely to contain hidden data. Chi-square steganalysis is a relatively simple and efficient method, and it can be effective against a wide range of steganography techniques. However, it is important to note that chi-square steganalysis is not infallible. Steganographers can develop techniques that are resistant to chi-square detection by exploiting the underlying statistical assumptions of the test. For example, steganographers can embed hidden messages in a way that does not significantly alter the distribution of pixel values in the image. Additionally, steganographers can use techniques to introduce noise into the image, which can make it more difficult for chi-square steganalysis to distinguish between a cover image and a stego image.

5.2.5 Fusion Technique

The fusion technique is a steganalysis method as a combination of well-known statistical steganalysis techniques. A fusion technique can be created according to different rules such as Min, Max, Mean, Median, and Product [Kharrazi et al., 2006]. In our tests, we used StegExpose's implementation of a Mean rule based classifier combination created with the statistical steganalysis method mentioned above [Boehm, 2014].

5.3 Visual Attack

The visual attack is a steganography method that is used to discover the existence of LSB embedding in images by making changes to the pixel values depending on their least significant bits. This process creates an output image that is visually different for a cover image and a stego image. The visual method is often neglected as a steganalysis test method because it is hard to automate [Marçal and Pereira, 2005] and can give unreliable results for images that are too noisy to distinguish. Visual attack on regular LSB embedding schemes uncovers obvious artifacts to visually detect [Westfeld and Pfitzmann, 2000]. For the steganography methods, even if they have randomized bit placement, using visually uniform places of an image for embedding can be detected by this type of steganalysis attack. This attack can be applied simply by an iteration over stego image's pixel values to modify the color channel values that ends with a "0" bit to a full "0" value and the channel values that ends with a "1" bit to a full "1" value. This will maximize and minimize the channel values creating a clear visible distinction between color channels which will expose patterns of LSB data hidden in the stego image visually.

6 Experimental Results

The proposed method is tested against Battlesteg and Filterfirst methods proposed in [Hempstalk, 2006] as well as the LSB embedding with no modification. The test methods include distortion measurements and statistical steganalysis methods mentioned above. Test images are also tested with additional visual attack [Westfeld and Pfitzmann, 2000] to uncover unique visual fingerprints of the aforementioned steganography methods.

The test data consists of 30 royalty-free images acquired from [Pexels, 2022] consisting of everyday objects and scenarios and various resolutions. Test images are embedded with the same 40 kilobytes of data with every steganography method tested.

The proposed algorithm uses a version of the DeeplabV3 image segmentation model [Chen et al., 2017] which uses the MobileNetV2 [Sandler et al.] backbone and pre-trained with the PASCAL VOC 2012 dataset [Everingham et al., 2010] for the test setup.

Distortion measurement tests are done with the scikit-image metrics [Van der Walt et al., 2014], and the statistical steganalysis tests are done with the StegExpose [Boehm, 2014] application. A Python script is used for the visual attack.

In the following tables, the Originals column refers to the images with no steganographic embedding. The LSB column refers to the Least Significant Bit method with no modification as explained in section 2, other columns on the tables are named after the steganography algorithms used.

6.1 Distortion Measurement Test Results

The following table shows the average distortion metric scores of stego images for each method as well as the reference original images with no steganography applied.

| Filename | Originals | Proposed Method | Battlesteg | LSB | FilterFirst |
|--------------|-----------|-----------------|------------|----------|-------------|
| apple.png | 0 | 0.027431 | 0.027352 | 0.027450 | 0.027399 |
| armchair.png | 0 | 0.021819 | 0.021904 | 0.022607 | 0.021762 |
| bird.png | 0 | 0.112272 | 0.112395 | 0.112644 | 0.112475 |
| bird2.png | 0 | 0.021865 | 0.021828 | 0.021933 | 0.021824 |
| birds.png | 0 | 0.021736 | 0.021795 | 0.021707 | 0.021659 |
| bunny.png | 0 | 0.087221 | 0.087479 | 0.087233 | 0.087301 |
| car.png | 0 | 0.049149 | 0.049052 | 0.049341 | 0.049240 |
| cat1.png | 0 | 0.021961 | 0.021933 | 0.021881 | 0.021883 |
| cat2.png | 0 | 0.044336 | 0.044446 | 0.044689 | 0.044534 |
| cat3.png | 0 | 0.049218 | 0.049243 | 0.049090 | 0.049218 |
| chair.png | 0 | 0.025926 | 0.025910 | 0.026044 | 0.025817 |
| cows.png | 0 | 0.025838 | 0.025791 | 0.025774 | 0.025879 |
| deers.png | 0 | 0.049162 | 0.049198 | 0.049109 | 0.049141 |
| dog1.png | 0 | 0.094703 | 0.094726 | 0.094429 | 0.094506 |
| dog2.png | 0 | 0.092210 | 0.092040 | 0.091980 | 0.091860 |
| dog3.png | 0 | 0.081116 | 0.081226 | 0.081531 | 0.080867 |
| dogs.png | 0 | 0.049346 | 0.049294 | 0.048975 | 0.049113 |
| duck.png | 0 | 0.055118 | 0.055235 | 0.055027 | 0.055088 |
| lion.png | 0 | 0.049007 | 0.049219 | 0.049280 | 0.049107 |

| | | | | | |
|---------------------|---|----------|----------|----------|----------|
| man1.png | 0 | 0.049027 | 0.049124 | 0.049254 | 0.049054 |
| man2.png | 0 | 0.021863 | 0.021880 | 0.021846 | 0.021921 |
| panda.png | 0 | 0.021857 | 0.021931 | 0.021986 | 0.021998 |
| plane.png | 0 | 0.025903 | 0.025889 | 0.025859 | 0.025815 |
| plane2.png | 0 | 0.019419 | 0.019386 | 0.019436 | 0.019462 |
| radio.png | 0 | 0.021860 | 0.021784 | 0.021875 | 0.021693 |
| squirrel.png | 0 | 0.037246 | 0.037354 | 0.037337 | 0.037315 |
| woman1.png | 0 | 0.049227 | 0.049208 | 0.049229 | 0.049244 |
| woman2.png | 0 | 0.050944 | 0.050814 | 0.051069 | 0.050953 |
| car2.png | 0 | 0.032713 | 0.032777 | 0.032777 | 0.032716 |
| motobike.png | 0 | 0.024509 | 0.024529 | 0.024904 | 0.024625 |

Table 1: MSE test results of multiple methods

Table 1 shows the Mean Squared Error (MSE) distortion metric results for the tested steganography methods. The results in the table show similar very close to zero scores for every method tested. As seen in the originals column that shows the original image's MSE value against itself, a closer score to zero means a lesser difference from the original image. As a result, this test did not show any significant visual distortion in the tested stego images.

| Filename | Originals | Proposed Method | Battlesteg | LSB | FilterFirst |
|---------------------|------------------|------------------------|-------------------|------------|--------------------|
| apple.png | inf | 63.748395 | 63.760895 | 63.745397 | 63.753531 |
| armchair.png | inf | 64.742486 | 64.725484 | 64.588397 | 64.753731 |
| bird.png | inf | 57.628089 | 57.623318 | 57.613737 | 57.620247 |
| bird2.png | inf | 64.733235 | 64.740569 | 64.719894 | 64.741487 |
| birds.png | inf | 64.759034 | 64.747244 | 64.764873 | 64.774406 |
| bunny.png | inf | 58.724586 | 58.711746 | 58.723992 | 58.720590 |
| car.png | inf | 61.215617 | 61.224231 | 61.198682 | 61.207586 |
| cat1.png | inf | 64.714379 | 64.719876 | 64.730107 | 64.729809 |
| cat2.png | inf | 61.663191 | 61.652503 | 61.628766 | 61.643859 |
| cat3.png | inf | 61.209606 | 61.207316 | 61.220881 | 61.209525 |
| chair.png | inf | 63.993461 | 63.996129 | 63.973703 | 64.011766 |
| cows.png | inf | 64.008188 | 64.016139 | 64.018956 | 64.001358 |
| deers.png | inf | 61.214538 | 61.211357 | 61.219207 | 61.216400 |
| dog1.png | inf | 58.367171 | 58.366093 | 58.379751 | 58.376211 |
| dog2.png | inf | 58.483034 | 58.491030 | 58.493860 | 58.499527 |
| dog3.png | inf | 59.039758 | 59.033834 | 59.017587 | 59.053088 |

| | | | | | |
|---------------------|-----|-----------|-----------|-----------|-----------|
| dogs.png | inf | 61.198279 | 61.202822 | 61.231021 | 61.218802 |
| duck.png | inf | 60.717879 | 60.708628 | 60.725013 | 60.720229 |
| lion.png | inf | 61.228261 | 61.209498 | 61.204087 | 61.219369 |
| man1.png | inf | 61.226412 | 61.217905 | 61.206400 | 61.224088 |
| man2.png | inf | 64.733801 | 64.730381 | 64.737009 | 64.722258 |
| panda.png | inf | 64.734975 | 64.720170 | 64.709287 | 64.706920 |
| plane.png | inf | 63.997340 | 63.999632 | 64.004650 | 64.012162 |
| plane2.png | inf | 65.248507 | 65.255928 | 65.244707 | 65.238838 |
| radio.png | inf | 64.734232 | 64.749456 | 64.731242 | 64.767693 |
| squirrel.png | inf | 62.420000 | 62.407440 | 62.409459 | 62.411937 |
| woman1.png | inf | 61.208744 | 61.210441 | 61.208609 | 61.207262 |
| woman2.png | inf | 61.059855 | 61.071010 | 61.049240 | 61.059100 |
| car2.png | inf | 62.983544 | 62.975125 | 62.975125 | 62.983166 |
| motobike.png | inf | 64.237563 | 64.233969 | 64.168132 | 64.216985 |

Table 2: PSNR test results for multiple methods

Table 2 shows us the Peak Signal to Noise Ratio (PSNR) test values of the tested steganography metrics. As seen in section 5 of this study, this metric is derived from the MSE metric and it gives correlated results. In contrast to the MSE the higher value here means less distortion from the original image. All of the steganography techniques tested resulted in a high enough PSNR score that a visual distortion is not present in the image.

| Filename | Originals | Proposed Method | BattleSteg | LSB | Filterfirst |
|---------------------|------------------|------------------------|-------------------|------------|--------------------|
| apple.png | 1 | 0.999842 | 0.999740 | 0.999604 | 0.999885 |
| armchair.png | 1 | 0.999923 | 0.999929 | 0.999786 | 0.999980 |
| bird.png | 1 | 0.999626 | 0.999134 | 0.998340 | 0.999535 |
| bird2.png | 1 | 0.999793 | 0.999868 | 0.999721 | 0.999990 |
| birds.png | 1 | 0.999958 | 0.999890 | 0.999681 | 0.999994 |
| bunny.png | 1 | 0.999454 | 0.999447 | 0.998960 | 0.999621 |
| car.png | 1 | 0.999894 | 0.999830 | 0.999860 | 0.999970 |
| cat1.png | 1 | 0.999879 | 0.999841 | 0.999792 | 0.999968 |
| cat2.png | 1 | 0.999776 | 0.999557 | 0.999361 | 0.999771 |
| cat3.png | 1 | 0.999909 | 0.999842 | 0.999579 | 0.999946 |
| chair.png | 1 | 0.999876 | 0.999873 | 0.999681 | 0.999958 |
| cows.png | 1 | 0.999737 | 0.999807 | 0.999758 | 0.999919 |
| deers.png | 1 | 0.999891 | 0.999771 | 0.999235 | 0.999934 |

| | | | | | |
|---------------------|---|----------|----------|----------|----------|
| dog1.png | 1 | 0.999343 | 0.999036 | 0.999004 | 0.999315 |
| dog2.png | 1 | 0.999069 | 0.998965 | 0.998768 | 0.999227 |
| dog3.png | 1 | 0.999507 | 0.999470 | 0.998841 | 0.999719 |
| dogs.png | 1 | 0.999721 | 0.999682 | 0.998994 | 0.999889 |
| duck.png | 1 | 0.999400 | 0.999633 | 0.999277 | 0.999833 |
| lion.png | 1 | 0.999838 | 0.999804 | 0.999252 | 0.999949 |
| man1.png | 1 | 0.999627 | 0.999443 | 0.999239 | 0.999681 |
| man2.png | 1 | 0.999896 | 0.999793 | 0.999703 | 0.999885 |
| panda.png | 1 | 0.999742 | 0.999885 | 0.999821 | 0.999953 |
| plane.png | 1 | 0.999774 | 0.999875 | 0.999636 | 0.999986 |
| plane2.png | 1 | 0.999802 | 0.999749 | 0.999741 | 0.999838 |
| radio.png | 1 | 0.999745 | 0.999847 | 0.999688 | 0.999976 |
| squirrel.png | 1 | 0.999727 | 0.999829 | 0.999457 | 0.999967 |
| woman1.png | 1 | 0.999614 | 0.999477 | 0.999286 | 0.999646 |
| woman2.png | 1 | 0.999620 | 0.999449 | 0.999324 | 0.999679 |
| car2.png | 1 | 0.999764 | 0.999671 | 0.999503 | 0.999915 |
| motobike.png | 1 | 0.999754 | 0.999905 | 0.999847 | 0.999975 |

Table 3: SSIM test results for multiple methods

Table 3 shows the Structural Similarity Index (SSIM) scores of the tested methods. This method as discussed in section 5 focuses on the human visual perception difference between the two images. Scores for every steganography method are very close to the perfect score of 1, which tells us as in other distortion metrics no perceptible distortion is present between the original and the stego images.

| Method | Proposed Method | BattleSteg | LSB | FilterFirst | Original Images |
|---------------|------------------------|-------------------|--------------|--------------------|------------------------|
| MSE | 0.04446673333 | 0.0444914000 | 0.0445432000 | 0.04444896667 | 0 |
| SSIM | 0.99971670000 | 0.9996680667 | 0.9994246333 | 0.99983013330 | 1 |
| PSNR | 62.2658053300 | 62.264005630 | 62.254725700 | 62.2673976700 | inf |

Table 4: Average test results for distortion measurement tests



Figure 4: Sample unmodified image (Left) and the stego image output of the proposed method (Right)

Every steganography method tested in distortion metrics has shown acceptable ranges of distortion. For PSNR metric value ≥ 30 will not make a visible impact on the image [Hsiao et al., 2009]. For MSE and SSIM metrics no visible distortion is present as every steganography algorithm tested achieved close to perfect scores in both distortion measurement tests. Tested distortion metrics and the example images show that no perceptible distortion is present in the stego images.

6.2 Statistical Steganalysis Test Results

The following tables show the statistical steganalysis results of tested methods. Each percentage value represents the probability of hidden data for the steganalysis detector used.

| File name | Originals | Proposed Method | BattleSteg | LSB | FilterFirst |
|--------------|-----------|-----------------|------------|--------|-------------|
| apple.png | 2.15% | 2.72% | 5.09% | 6.57% | 1.96% |
| armchair.png | 3.19% | 4.19% | 5.65% | 12.56% | 3.59% |
| bird.png | 5.39% | 8.11% | 20.89% | 30.98% | 10.01% |
| bird2.png | 1.14% | 3.85% | 3.56% | 5.70% | 1.40% |
| birds.png | 0.73% | 1.05% | 2.72% | 5.24% | 0.71% |
| bunny.png | 0.82% | 10.11% | 12.24% | null | 9.76% |
| car.png | 6.69% | 9.49% | 12.89% | 9.17% | 6.78% |
| car2.png | 0.55% | 2.92% | 4.97% | 4.56% | 1.61% |
| cat1.png | 0.05% | 0.90% | 2.25% | 3.08% | 0.27% |
| cat2.png | 0.26% | 1.01% | 4.70% | 5.90% | 0.91% |
| cat3.png | 1.12% | 3.12% | 6.19% | 14.29% | 2.58% |
| chair.png | 0.63% | 2.28% | 2.98% | 6.70% | 1.27% |
| cows.png | 0.55% | 3.38% | 3.33% | 7.74% | 1.30% |
| deers.png | 2.27% | 3.28% | 6.68% | 15.30% | 2.42% |

| | | | | | |
|---------------------|-------|--------|--------|--------|-------|
| dog1.png | 0.09% | 3.46% | 12.04% | 16.84% | 4.76% |
| dog2.png | 0.10% | 5.24% | 12.17% | 14.83% | 6.25% |
| dog3.png | 0.12% | 2.51% | 8.79% | 26.24% | 1.46% |
| dogs.png | 1.47% | 5.00% | 6.84% | 16.17% | 2.83% |
| duck.png | 0.12% | 10.77% | 6.85% | 15.82% | 3.54% |
| lion.png | 0.03% | 2.15% | 4.63% | null | 1.47% |
| man1.png | 0.35% | 2.23% | 6.43% | 7.84% | 2.17% |
| man2.png | 1.78% | 2.42% | 3.90% | 2.33% | 0.47% |
| motobike.png | 1.21% | 2.92% | 4.97% | 3.22% | 3.07% |
| panda.png | 0.56% | 3.49% | 12.04% | 8.32% | 1.31% |
| plane.png | 0.32% | 2.28% | 2.50% | 3.94% | 0.35% |
| plane2.png | 1.92% | 3.46% | 2.98% | 12.25% | 2.23% |
| radio.png | 0.07% | 8.75% | 5.51% | 9.13% | 3.80% |
| squirrel.png | 0.29% | 4.19% | 5.65% | 7.87% | 1.82% |
| woman1.png | 0.55% | 0.90% | 2.25% | 4.56% | 1.61% |
| woman2.png | 3.14% | 3.38% | 3.33% | 8.58% | 3.84% |
| Average | 1.24% | 4.27% | 6.41% | 10.17% | 2.28% |

Table 5: Primary sets detector results for tested methods

Table 5 shows the test results for the Primary Sets steganalysis method results for the tested methods. The percentage values shows the probability of the existence of an embedded message in the image. With an average of 4.27% the proposed method is in second place of the 4 methods tested for this metric.

| File name | Originals | Proposed Method | BattleSteg | LSB | FilterFirst |
|---------------------|------------------|------------------------|-------------------|------------|--------------------|
| apple.png | 0.14% | 0.14% | 0.16% | 6.24% | 0.14% |
| armchair.png | 0.01% | 0.01% | 0.01% | 4.58% | 0.01% |
| bird.png | 16.32% | 16.32% | 20.07% | 24.99% | 20.31% |
| bird2.png | 0.11% | 0.11% | 0.12% | 5.20% | 0.11% |
| birds.png | 0.13% | 0.13% | 0.15% | 5.52% | 0.14% |
| bunny.png | 0.62% | 0.62% | 0.89% | 21.90% | 0.62% |
| car.png | 1.04% | 1.04% | 1.44% | 11.20% | 1.62% |
| car2.png | 0.08% | 0.08% | 0.08% | 7.93% | 0.08% |
| cat1.png | 0.01% | 0.01% | 0.01% | 4.74% | 0.01% |
| cat2.png | 0.14% | 0.14% | 0.14% | 10.18% | 0.14% |
| cat3.png | 0.75% | 1.43% | 0.86% | 14.44% | 0.74% |

| | | | | | |
|---------------------|-------|-------|-------|--------|-------|
| chair.png | 0.05% | 0.05% | 0.05% | 6.07% | 0.05% |
| cows.png | 0.00% | 0.00% | 0.00% | 5.42% | 0.00% |
| deers.png | 0.11% | 0.11% | 0.11% | 11.05% | 0.10% |
| dog1.png | 0.05% | 0.05% | 0.06% | 19.88% | 0.05% |
| dog2.png | 0.17% | 0.17% | 0.20% | 19.37% | 0.18% |
| dog3.png | 0.10% | 0.10% | 0.11% | 16.71% | 0.10% |
| dogs.png | 0.49% | 0.49% | 0.50% | 13.03% | 0.48% |
| duck.png | 1.72% | 1.72% | 2.26% | 20.07% | 1.78% |
| lion.png | 0.37% | 0.37% | 0.41% | 12.32% | 0.37% |
| man1.png | 0.46% | 0.46% | 0.49% | 10.59% | 0.46% |
| man2.png | 0.72% | 0.72% | 0.77% | 5.56% | 0.73% |
| motobike.png | 0.02% | 0.02% | 0.02% | 5.26% | 0.02% |
| panda.png | 2.08% | 2.08% | 2.23% | 20.77% | 2.12% |
| plane.png | 0.10% | 0.10% | 0.10% | 5.97% | 0.10% |
| plane2.png | 0.16% | 0.16% | 0.18% | 4.86% | 0.18% |
| radio.png | 0.06% | 0.06% | 0.06% | 4.67% | 0.06% |
| squirrel.png | 0.09% | 0.09% | 0.10% | 8.93% | 0.09% |
| woman1.png | 0.63% | 0.63% | 0.88% | 13.28% | 0.65% |
| woman2.png | 0.09% | 0.09% | 0.10% | 11.03% | 0.09% |
| Average | 0.91% | 0.93% | 1.11% | 11.08% | 1.04% |

Table 6: Chi-Square detector results for tested methods

Table 6 shows the Chi-Square analysis results for the 4 tested steganography methods. As the other statistical steganalysis results the percentage of probability is given for the existence of a secret message embedding in the image. With a 0.93% average, the proposed method is placed first for this test metric.

| File name | Originals | Proposed Method | BattleSteg | LSB | FilterFirst |
|---------------------|------------------|------------------------|-------------------|------------|--------------------|
| apple.png | 5.66% | 5.99% | 8.43% | 8.99% | 5.66% |
| armchair.png | 4.70% | 5.29% | 6.64% | 11.22% | 5.00% |
| bird.png | 3.28% | 6.63% | 17.94% | 21.55% | 8.11% |
| bird2.png | 3.14% | 5.65% | 5.25% | 7.00% | 2.96% |
| birds.png | 1.49% | 1.73% | 3.43% | 5.08% | 0.95% |
| bunny.png | 0.14% | 9.43% | 12.20% | 20.63% | 8.65% |
| car.png | 6.19% | 8.96% | 11.02% | 8.80% | 6.00% |
| car2.png | 0.57% | 2.84% | 4.81% | 4.27% | 1.37% |

| | | | | | |
|--------------|-------|-------|--------|--------|-------|
| cat1.png | 0.03% | 0.68% | 2.08% | 2.51% | 0.18% |
| cat2.png | 0.08% | 1.02% | 4.76% | 5.06% | 1.06% |
| cat3.png | 2.14% | 4.34% | 7.22% | 12.68% | 3.28% |
| chair.png | 1.62% | 4.10% | 4.17% | 8.35% | 1.77% |
| cows.png | 1.54% | 4.26% | 4.24% | 7.51% | 2.01% |
| deers.png | 1.32% | 2.20% | 5.89% | 11.69% | 1.91% |
| dog1.png | 0.13% | 3.87% | 12.06% | 13.21% | 5.47% |
| dog2.png | 0.23% | 5.80% | 12.47% | 11.84% | 6.72% |
| dog3.png | 0.33% | 3.67% | 9.19% | 19.75% | 2.06% |
| dogs.png | 1.24% | 4.20% | 6.85% | 12.23% | 2.86% |
| duck.png | 0.30% | 9.96% | 7.25% | 12.43% | 3.74% |
| lion.png | 0.96% | 3.49% | 5.87% | 18.96% | 2.04% |
| man1.png | 0.20% | 1.97% | 6.05% | 8.05% | 1.73% |
| man2.png | 0.06% | 1.05% | 2.55% | 2.58% | 1.12% |
| motobike.png | 1.95% | 7.46% | 4.63% | 6.45% | 2.82% |
| panda.png | 0.09% | 7.44% | 3.04% | 3.18% | 2.12% |
| plane.png | 1.78% | 5.68% | 4.40% | 7.88% | 1.70% |
| plane2.png | 0.72% | 3.89% | 3.81% | 3.00% | 2.98% |
| radio.png | 0.69% | 3.76% | 2.91% | 3.82% | 0.61% |
| squirrel.png | 2.47% | 5.83% | 6.06% | 11.03% | 2.67% |
| woman1.png | 0.01% | 3.06% | 6.66% | 7.52% | 4.33% |
| woman2.png | 0.75% | 2.85% | 6.56% | 6.91% | 2.25% |
| Average | 1.46% | 4.57% | 6.61% | 9.47% | 3.14% |

Table 7: Sample Pairs detector results for tested methods

Table 7 shows the results for the Sample Pairs steganalysis method results as probability like other statistical test results. With the 4.57% average, the proposed method is the second best among the tested steganography methods according to this metric.

| File name | Originals | Proposed Method | BattleSteg | LSB | FilterFirst |
|--------------|-----------|-----------------|------------|--------|-------------|
| apple.png | 6.16% | 6.59% | 9.03% | 9.66% | 6.21% |
| armchair.png | 5.35% | 6.05% | 7.34% | 11.61% | 5.81% |
| bird.png | 3.13% | 6.93% | 17.72% | 21.43% | 8.78% |
| bird2.png | 3.55% | 6.37% | 5.72% | 7.27% | 3.31% |
| birds.png | 1.89% | 2.21% | 3.87% | 5.63% | 1.24% |

| | | | | | |
|---------------------|-------|--------|--------|--------|-------|
| bunny.png | 0.19% | 10.19% | 12.68% | 19.38% | 9.58% |
| car.png | 6.22% | 9.01% | 11.03% | 8.94% | 5.89% |
| car2.png | 0.67% | 3.31% | 5.13% | 4.40% | 1.80% |
| cat1.png | 0.14% | 0.95% | 2.21% | 2.51% | 0.34% |
| cat2.png | 0.08% | 1.36% | 4.97% | 5.18% | 1.47% |
| cat3.png | 2.19% | 4.50% | 7.15% | 12.91% | 3.45% |
| chair.png | 2.05% | 4.65% | 4.60% | 7.98% | 2.27% |
| cows.png | 1.80% | 4.80% | 4.62% | 7.31% | 2.48% |
| deers.png | 1.22% | 2.51% | 5.85% | 11.07% | 1.94% |
| dog1.png | 0.10% | 4.83% | 12.51% | 12.96% | 6.74% |
| dog2.png | 0.36% | 6.82% | 13.03% | 11.79% | 7.98% |
| dog3.png | 0.31% | 4.42% | 8.84% | 18.85% | 2.46% |
| dogs.png | 1.17% | 4.69% | 6.79% | 11.34% | 3.16% |
| duck.png | 0.52% | 10.46% | 7.83% | 12.01% | 4.49% |
| lion.png | 1.45% | 4.07% | 6.33% | 17.67% | 2.63% |
| man1.png | 0.07% | 2.45% | 6.30% | 7.78% | 2.38% |
| man2.png | 0.07% | 1.35% | 2.75% | 2.88% | 1.45% |
| motobike.png | 1.76% | 7.44% | 4.35% | 5.88% | 2.73% |
| panda.png | 0.32% | 6.94% | 3.03% | 3.18% | 2.30% |
| plane.png | 2.21% | 6.16% | 4.86% | 8.15% | 2.07% |
| plane2.png | 0.60% | 3.67% | 3.86% | 2.96% | 3.22% |
| radio.png | 0.88% | 4.16% | 3.13% | 4.03% | 0.77% |
| squirrel.png | 2.29% | 5.98% | 5.91% | 10.49% | 2.58% |
| woman1.png | 0.08% | 3.53% | 7.08% | 7.28% | 4.98% |
| woman2.png | 0.71% | 3.21% | 6.75% | 6.73% | 2.64% |
| Average | 1.58% | 4.99% | 6.84% | 9.31% | 3.57% |

Table 8: RS Analysis detector results for tested methods

Table 8 shows the RS analysis test results for the steganalysis methods tested. With a 4.99% average score, the proposed method is the second best among the tested methods for this metric.

| File name | Originals | Proposed Method | BattleSteg | LSB | FilterFirst |
|---------------------|------------------|------------------------|-------------------|------------|--------------------|
| apple.png | 3.53% | 3.86% | 5.68% | 7.87% | 3.49% |
| armchair.png | 3.32% | 3.89% | 4.91% | 9.99% | 3.60% |
| bird.png | 8.28% | 10.46% | 19.56% | 25.80% | 13.03% |

| | | | | | |
|---------------------|-------|-------|-------|--------|-------|
| bird2.png | 1.99% | 3.99% | 3.66% | 6.29% | 1.95% |
| birds.png | 1.06% | 1.28% | 2.54% | 5.37% | 0.76% |
| bunny.png | 0.44% | 7.59% | 9.50% | 20.64% | 7.16% |
| car.png | 4.65% | 6.51% | 8.45% | 9.77% | 4.76% |
| car2.png | 0.47% | 2.29% | 3.75% | 5.29% | 1.21% |
| cat1.png | 0.07% | 0.62% | 1.49% | 3.44% | 0.20% |
| cat2.png | 0.14% | 0.88% | 3.64% | 6.58% | 0.89% |
| cat3.png | 1.35% | 3.02% | 4.73% | 13.88% | 2.26% |
| chair.png | 0.91% | 2.33% | 2.55% | 6.92% | 1.20% |
| cows.png | 0.97% | 3.11% | 3.05% | 7.00% | 1.45% |
| deers.png | 1.20% | 1.97% | 4.21% | 12.47% | 1.49% |
| dog1.png | 0.09% | 3.05% | 9.17% | 15.72% | 4.25% |
| dog2.png | 0.22% | 4.51% | 9.47% | 14.46% | 5.28% |
| dog3.png | 0.17% | 2.34% | 5.91% | 20.60% | 1.34% |
| dogs.png | 1.05% | 3.40% | 4.71% | 13.51% | 2.16% |
| duck.png | 0.67% | 8.23% | 6.05% | 15.08% | 3.39% |
| lion.png | 0.61% | 2.20% | 3.79% | 14.99% | 1.49% |
| man1.png | 0.29% | 1.71% | 4.40% | 8.74% | 1.67% |
| man2.png | 0.28% | 1.15% | 2.26% | 3.64% | 1.20% |
| motobike.png | 1.64% | 5.40% | 3.29% | 6.57% | 2.20% |
| panda.png | 1.07% | 5.76% | 2.41% | 7.36% | 1.75% |
| plane.png | 1.32% | 4.25% | 3.32% | 7.58% | 1.29% |
| plane2.png | 0.51% | 2.97% | 2.93% | 3.51% | 2.36% |
| radio.png | 0.49% | 2.87% | 2.15% | 4.12% | 0.44% |
| squirrel.png | 1.43% | 3.85% | 3.89% | 10.56% | 1.64% |
| woman1.png | 0.26% | 2.36% | 4.85% | 9.90% | 3.14% |
| woman2.png | 0.36% | 1.91% | 4.39% | 8.54% | 1.52% |
| Average | 1.29% | 3.59% | 5.02% | 10.21% | 2.62% |

Table 9: Fusion (Mean) detector results for tested methods

Table 9 shows the Fusion detector using the mean rule, as discussed in section 5 this steganalysis metric is created by fusing multiple different statistical steganalysis metrics. The performance of the proposed method is 3.59% detection on average in tested images. This places the proposed algorithm as the second best among the tested steganography methods in this study.

| Test Method | Originals | Proposed M. | BattleSteg | LSB | FilterFirst |
|---------------------|-----------|-------------|------------|--------|-------------|
| Primary Sets | 1.25% | 4.27% | 6.35% | 10.17% | 2.85% |
| Chi-Square | 0.89% | 0.93% | 1.09% | 11.06% | 1.05% |
| Sample Pairs | 1.46% | 4.57% | 6.61% | 9.47% | 3.14% |

| | | | | | |
|---------------------|-------|-------|-------|--------|-------|
| RS Analysis | 1.58% | 4.99% | 6.84% | 9.31% | 3.57% |
| Fusion(Mean) | 1.29% | 3.59% | 5.02% | 10.21% | 2.62% |

Table 10: Average statistical steganalysis results for tested methods

In statistical steganalysis tests conducted on the aforementioned samples, on average the proposed method shows better performance than the LSB and the BattleSteg [Hempstalk, 2006] methods. However, the proposed method did not achieve better average results than the FilterFirst [Hempstalk, 2006] method.

6.3 Visual Attack Test Results

The following images are sample test results of applied visual attacks on tested steganography methods.

In Figure 5 visual attack results of steganography methods are separated using the following letters:

- A : Original test image with no visual attack applied.
- B : Original test image visual attack applied.
- C : Image embedded with proposed method.
- D : Image embedded with BattleSteg method.
- E : Image embedded with FilterFirst method.
- F : Image embedded with LSB method.

All visual attack test results can be seen hosted at full quality in the referenced dataset [Demircan, 2023b].

The code of the proposed method and the dataset used for the tests can be obtained from the referenced public repository [Demircan, 2023a].

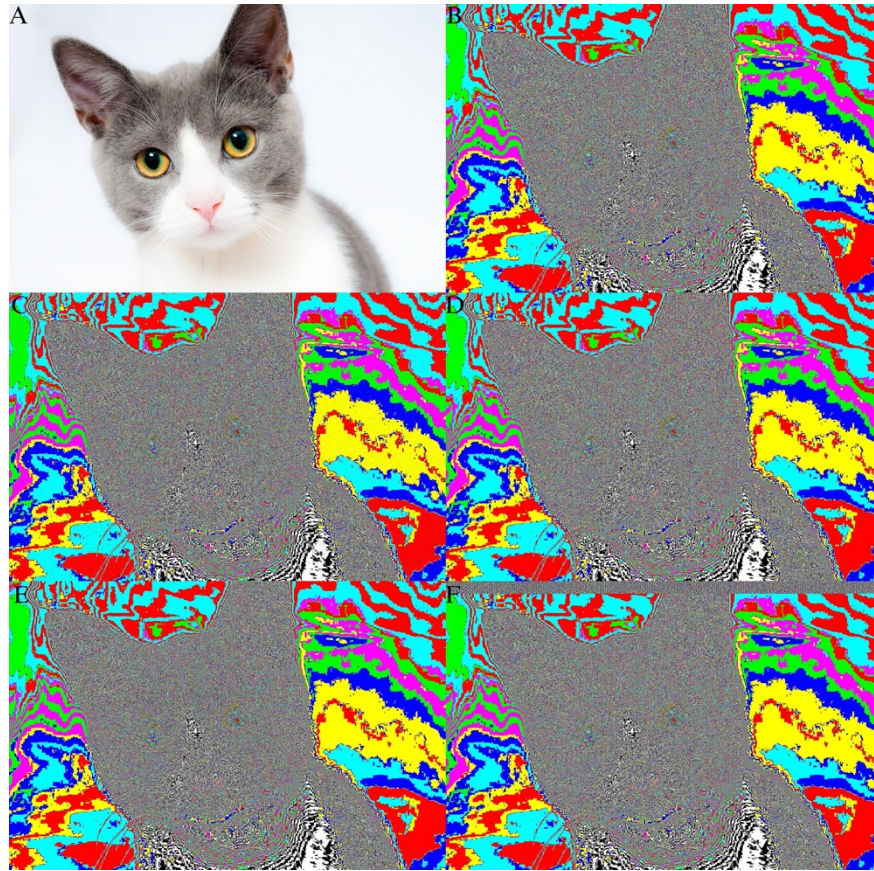


Figure 5: Sample visual attack results of tested methods

7 Conclusion and Future Work

The idea of an artificial intelligence image segmentation model is used in this research to provide a secure LSB method for image steganography. The proposed method is tested against other methods in the literature for performance analysis. The visual attack results of the proposed method achieved substantial improvement over the LSB method with no modifications and have overall good resistance to this attack. While the BattleSteg [Hempstalk, 2006] and LSB methods both have shown easily recognizable visual signatures, the proposed method and the FilterFirst [Hempstalk, 2006] method did not show recognizable visual signatures. As for the statistical steganalysis methods, Chi-Square [Westfeld and Pfitzmann, 2000] analysis resulted in the worst detection on average for the proposed method. The Primary Sets [Dumitrescu, Wu and Memon, 2002] analysis, Sample Pairs [Dumitrescu, Wu and Wang, 2003] analysis, RS analysis [Fridrich, Goljan and Du, 2001], and the Mean Fusion [Kharrazi, Sencar and Memon, 2006] detector results shown that the proposed method is the second least detectable among tested methods. In summary, the proposed method's statistical steganalysis

scores are on average better than other methods tested except one. As for the distortion measurement tests show, none of the tested steganography methods created enough distortion to be perceived by the human eye, improvement can be achieved on both statistical and visual attack [Westfeld and Pfitzmann, 2000] aspects with new pixel distribution techniques.

Future work includes testing the reliability of the proposed algorithm with multiple other image segmentation neural networks and training datasets as well as other image formats.

References

- [Abuzanouneh and Hadwan, 2021] Abuzanouneh, K.I.M. AND Hadwan, M. 2021 : Multi-Stage Protection using Pixel Selection Technique for Enhancing Steganography. *International Journal of Communication Networks and Information Security* 13, 55-61.
- [Al-Ahmad, Almousa and Abuein, 2021] Al-Ahmad, A., Almousa, O.S. AND Abuein, Q. 2021 : Enhancing steganography by image segmentation and multi-level deep hiding. *International Journal of Communication Networks and Information Security* 13, 143-150.
- [Alyousuf, Din and Qasim, 2020] Alyousuf, F.Q.A., Din, R. AND Qasim, A.J. 2020 : Analysis review on spatial and transform domain technique in digital steganography. *Bulletin of Electrical Engineering and Informatics* 9, 573-581.
- [Ansari, Mohammadi and Parvez, 2019] Ansari, A.S., Mohammadi, M.S. AND Parvez, M.T. 2019 : A comparative study of recent steganography techniques for multiple image formats. *International Journal of Computer Network and Information Security* 11, 11-25.
- [Arya and Soni, 2018] Arya, A. AND Soni, S. 2018 : A literature review on various recent steganography techniques. *International Journal on Future Revolution in Computer Science & Communication Engineering* 4, 143-149.
- [Bawaneh and Obeidat, 2016] Bawaneh, M.J. AND Obeidat, A.A. 2016 : A secure robust gray scale image steganography using image segmentation. *Journal of Information Security* 7, 152.
- [Boehm, 2014] Boehm, B. 2014 : Stegexpose-A tool for detecting LSB steganography. *arXiv preprint arXiv:1410.6656*.
- [Cheddad, Condell, Curran and Mc Kevitt, 2010] Cheddad, A., Condell, J., Curran, K. AND Mc Kevitt, P. 2010 : Digital image steganography: Survey and analysis of current methods. *Signal Processing* 90, 727-752.
- [Chen, Papandreou, Schroff and Adam, 2017] Chen, L.-C., Papandreou, G., Schroff, F. AND Adam, H. 2017 : Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.
- [Das and Tuithung, 2012] Das, R. AND Tuithung, T. 2012 : A novel steganography method for image based on Huffman Encoding. In *2012 3rd National Conference on Emerging Trends and Applications in Computer Science IEEE*, 14-18.
- [Demircan, 2023a] Demircan, Y.Y. 2023a : Code And Dataset of The Proposed Method *github*. <https://github.com/yasirdemircan/SegmentationSteganography>
- [Demircan, 2023b] Demircan, Y.Y. 2023b : Visual Attack Results Dataset *figshare*.10.6084/M9.FIGSHARE.22714303.
<http://dx.doi.org/10.6084/M9.FIGSHARE.22714303>

- [Duan, Jia, Li, Guo, Zhang and Qin, 2019] Duan, X., Jia, K., Li, B., Guo, D., Zhang, E. AND Qin, C. 2019 : Reversible image steganography scheme based on a U-Net structure. IEEE Access 7, 9314-9323.
- [Dumitrescu, Wu and Memon, 2002] Dumitrescu, S., Wu, X. AND Memon, N. 2002 : On steganalysis of random LSB embedding in continuous-tone images. In Proceedings. International conference on image processing IEEE, 641-644.
- [Dumitrescu, Wu and Wang, 2003] Dumitrescu, S., Wu, X. AND Wang, Z. 2003 : Detection of LSB steganography via sample pair analysis. In International workshop on information hiding Springer, 355-372.
- [Everingham, Van Gool, Williams, Winn and Zisserman, 2010] Everingham, M., Van Gool, L., Williams, C.K., Winn, J. AND Zisserman, A. 2010 : The pascal visual object classes (voc) challenge. International journal of computer vision 88, 303-338.
- [Fridrich, Goljan and Du, 2001] Fridrich, J., Goljan, M. AND Du, R. 2001 : Detecting LSB steganography in color, and gray-scale images. IEEE MultiMedia 8, 22-28.
- [Hariri, Karimi and Nosrati, 2011] Hariri, M., Karimi, R. AND Nosrati, M. 2011 : An introduction to steganography methods. World Applied Programming 1, 191-195.
- [Hempstalk, 2006] Hempstalk, K. 2006 : Hiding behind corners: Using edges in images for better steganography. In Proceedings of the Computing Women's Congress, Hamilton, New Zealand, 11-19.
- [Hogg, 1957] Hogg, R.V. 1957 : Introduction to Statistical Analysis JSTOR.
- [Hsiao, Chan and Chang, 2009] Hsiao, J.-Y., Chan, K.-F. AND Chang, J.M. 2009 : Block-based reversible data embedding. Signal Processing 89, 556-569.
- [Johnson and Jajodia, 1998] Johnson, N.F. AND Jajodia, S. 1998 : Steganalysis of images created using current steganography software. In International Workshop on Information Hiding Springer, 273-289.
- [Karampidis, Kavallieratou and Papadourakis, 2018] Karampidis, K., Kavallieratou, E. AND Papadourakis, G. 2018 : A review of image steganalysis techniques for digital forensics. Journal of information security and applications 40, 217-235.
- [Kaur and Kaur, 2014] Kaur, D. AND Kaur, Y. 2014 : Various image segmentation techniques: a review. International Journal of Computer Science and Mobile Computing 3, 809-814.
- [Khari, Garg, Gandomi, Gupta, Patan and Balusamy, 2020] Khari, M., Garg, A.K., Gandomi, A.H., Gupta, R., Patan, R. AND Balusamy, B. 2020 : Securing Data in Internet of Things (IoT) Using Cryptography and Steganography Techniques. IEEE Transactions on Systems, Man, and Cybernetics: Systems 50, 73-80.
- [Kharrazi, Sencar and Memon, 2006] Kharrazi, M., Sencar, H.T. AND Memon, N. 2006 : Improving steganalysis by fusion techniques: A case study with image steganography. In Transactions on Data Hiding and Multimedia Security I Springer, 123-137.
- [Luo, Qin, Xiang and Tan, 2020] Luo, Y., Qin, J., Xiang, X. AND Tan, Y. 2020 : Coverless image steganography based on multi-object recognition. IEEE Transactions on Circuits and Systems for Video Technology 31, 2779-2791.
- [Marçal and Pereira, 2005] Marçal, A.R. AND Pereira, P.R. 2005 : A steganographic method for digital images robust to RS steganalysis. In Image Analysis and Recognition: Second International Conference, ICIAR 2005, Toronto, Canada, September 28-30, 2005. Proceedings 2 Springer, 1192-1199.

[Mishra and Bhanodiya, 2015] Mishra, R. AND Bhanodiya, P. 2015 : A review on steganography and cryptography. In 2015 International Conference on Advances in Computer Engineering and Applications, 119-122.

[Nissar and Mir, 2010] Nissar, A. AND Mir, A.H. 2010 : Classification of steganalysis techniques: A study. Digital Signal Processing 20, 1758-1770.

[Nosrati, Hanani and Karimi, 2015] Nosrati, M., Hanani, A. AND Karimi, R. 2015 : Steganography in Image Segments Using Genetic Algorithm. In 2015 Fifth International Conference on Advanced Computing & Communication Technologies, 102-107.

[Pexels, 2022] Pexels 2022 : Free Stock Photos, Royalty Free Stock Images & Copyright Free Pictures.

[Pradhan, Sahu, Swain and Sekhar, 2016] Pradhan, A., Sahu, A.K., Swain, G. AND Sekhar, K.R. 2016 : Performance evaluation parameters of image steganography techniques. In 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS) IEEE, 1-8.

[Rachmawati, Tarigan and Ginting, 2018] Rachmawati, D., Tarigan, J. AND Ginting, A. 2018 : A comparative study of Message Digest 5 (MD5) and SHA256 algorithm. In Journal of Physics: Conference Series IOP Publishing, 012116.

[Sandler, Howard, Zhu, Zhmoginov and Chen, 2018] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. AND Chen, L.-C. : MobileNetV2: Inverted Residuals and Linear Bottlenecks.

[Sharma and Kumar, 2015] Sharma, S. AND Kumar, U. 2015 : Review of transform domain techniques for image steganography. International Journal of Science and Research 2, 1.

[Tiwari and Shandilya, 2010] Tiwari, N. AND Shandilya, D.M. 2010 : Evaluation of various LSB based methods of image steganography on GIF file format. International Journal of Computer Applications 6, 1-4.

[Van der Walt, Schönberger, Nunez-Iglesias, Boulogne, Warner, Yager, Gouillart and Yu, 2014] Van Der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E. AND Yu, T. 2014 : scikit-image: image processing in Python. PeerJ 2, e453.

[Wang, Bovik, Sheikh and Simoncelli, 2004] Wang, Z., Bovik, A.C., Sheikh, H.R. AND Simoncelli, E.P. 2004 : Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, 600-612.

[Westfeld and Pfitzmann, 2000] Westfeld, A. AND Pfitzmann, A. 2000: Attacks on steganographic systems. In International workshop on information hiding Springer, 61-76.

[Yang, Zhang and Yu, 2015] Yang, Y., Zhang, W. AND Yu, N. 2015: Improving Visual Quality of Reversible Data Hiding in Medical Image with Texture Area Contrast Enhancement. In 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 81-84.

[Yuheng and Hao, 2017] Yuheng, S. AND Hao, Y. 2017: Image segmentation algorithms overview. arXiv preprint arXiv:1707.02051.

[Zaini, 2021] Zaini, H.G. 2021: Image Segmentation to Secure LSB2 Data Steganography. Engineering, Technology & Applied Science Research 11, 6632-6636.