



LSB steganography detection in monochromatic still images using artificial neural networks

Julián D. Miranda¹ · Diego J. Parada¹

Received: 25 June 2020 / Revised: 30 March 2021 / Accepted: 29 August 2021 /

Published online: 17 September 2021

© The Author(s) 2021

Abstract

Embedding graphic content in multimedia through steganography is a useful and fast practice to hide information. However, detecting the use of this technique is complex and sometimes unsuccessful because variations are not visually perceptible. This article proposes the use of a binary classification model based on artificial neural networks to detect the presence of LSB steganography on monochromatic still images of 256x256 and 8 bits, based on the Standford Genome Project. The steganograms were generated by varying the payload from 0.1 to 0.5 to obtain image pairs of carriers and steganograms. For each steganogram, the following features were extracted from image histograms: kurtosis, skewness, standard deviation, range, median, harmonic mean, Hjorth mobility, and complexity. The results show that the classifier reaches a 91.45% accuracy in detecting LSB steganography when learning from all payloads, as well as a 96.78% individual classification accuracy in the best case with a payload of 0.5.

Keywords Steganography · Steganalysis · Artificial neural networks · Least significant bit

1 Introduction

The mechanism of human vision is based on the trichromatic identification (three-dimensional vision) of colors. These are perceived by photoreceptors (light-sensitive receivers) distributed on the periphery and interior of the fovea, a structure located in the retina in which the vision of objects triggers [7]. The information obtained, produced by this perception, is transduced from biological signals to electrical impulses that are sent through the optic nerve or second cranial nerve and interpreted by the brain [23], which is able to distinguish approximately 2.7 million different hues. This without

✉ Julián D. Miranda
julian.miranda@upb.edu.co

Diego J. Parada
diego.parada@upb.edu.co

¹ Faculty of Systems and Informatics Engineering, Pontifical Bolivarian University, Bucaramanga, Colombia

considering changes in color hues caused by variations in lighting conditions that do not affect color but perception [16].

Computational visual representation of the objects observed can be made by diverse visual multimedia, such as still images or photographs that have no movement. This representation is made in a matrix way in which delimited intensities are stored defined by a specific interval, usually containing 256 levels (8 bits) per dimension. If a trichromatic vision is considered, these 256 levels represented in three dimensions become more than 16 million chromatic possibilities [10]. Accordingly, there are more chances of representing object hues in still images than those the human eye can perceive, which is take advantage of by certain computational techniques focused on information concealment, such as steganography images.

Steganography is a practice to embed messages in innocuous objects carriers that are often framed in visual or auditory media of multimedia type [8]. When it comes to media as still images, there are two groups of stenographic techniques which allow concealing visual or written information (images or text messages either formatted or unformatted): LSB steganography by substitution and LSB steganography by coincidence or matching [28], where LSB stands for Least Significant Bit. In both cases, the carrier image is modified in the spatial domain, making variations in the least significant bits of the intensities in order to embed the message. However, although the image has changed its intensity hues, outwardly, it remains invariable to the human eye as a result of trichromatic perception explained.

Thus, steganography in still images proves to be hardly recognizable by the human eye, which has made it usable to conceal sensitive information through various transmission information media, breaking through human filters undetected [31] and expanding security breaches to control the static visual multimedia. LSB steganography is one of the most popular spatial steganography techniques implemented [31], as it is a relatively simple practice to perform and difficult to detect. There are other more complex spatial practices, such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). However, although they are more robust to modifications, they have limitations in the embedding capacity. If this capacity is increased, then the peak-signal-to-noise-ratio (PSNR), which measures the quality of the steganogram when compared to the carrier image, is considerably poor. In this way, LSB steganography becomes a widely used attack vector due to its vast embedding capacity and PSNR performance.

According to Chet Hosmer [20], President and CEO of Wetstone Technologies, approximately 0.6% of the content of images that are shared on websites have embedded graphic content. GeoEdge PR Department [9], provider of security solutions and verification of mobile and online advertising, in their report published on November 15, 2018, they document an exponential increase in the number of advertising images with steganography. According to GeoEdge, this rise has left a balance of approximately 120 million dollars in additional costs to advertisers and 920 million dollars to online advertising consumers in 2018, as most images contain malicious software embedded which executes automatically when opening them.

As David Buchanan [5] demonstrated when concealing information in images and publishing them on social networks like Twitter and Facebook, this information becomes undetectable both to the human eye and web services, making it undistinguishable between images with or without embedded content. This is because, although there are mechanisms for the detection of steganography in images, they are based on visual techniques, and statistics, which consider the purely spatial content [3], leading to

information concealing by means of techniques with random patterns and subtle variations that are visually undetectable.

The aim of this paper is to present the development of an algorithm for the detection of LSB steganography embedded in the spatial domain of monochromatic still images, by means of the use of binary classification models based on artificial neural networks, collaborating with steganalysis processes of one-dimensional images.

2 State of the art

A study of the state-of-the-art was carried out in order to identify various applications of Neural Networks in the identification of LSB steganography in still images. In Table 1, has been detailed the state of the art relevant to this research. This table shows that several works have been developed in the field of steganalysis of monochromatic images and color images, by using artificial neural networks and convolutional neural networks. The period analyzed in this study dates from 2016 to 2019, the interval in which this type of learning techniques for steganography detection has begun to be used. Several works of relevance for this research are detailed below.

Ingale et al. [14] in 2016 developed a model of artificial neural network for the steganalysis of LSB steganography in images with three spatial resolutions: 128x128, 256x256 and 512x512, and four densities of embedded content (payload): 5%, 10%, 15% and 20% of the total size of the input image. 150 input images were used for the training process and 50 images were used for testing the model, divided into 50% for carrier images and 50% for images with steganography. These images were generated by using steganPEG and Quickstego in JPEG format. Each image was sectioned in windows with a spatial resolution of 8 rows and 8 columns, extracting the histogram of each window and calculating entropy as the main attribute of model input, for a total of 16, 32 and 64 input attributes for images with spatial resolutions stipulated, respectively. The binary classifier application resulting from the neural network classification model achieved an accuracy of between 97% and 99%.

In 2016, Qian et al. [26], on the other hand, developed a learning model based on Convolutional Neural Networks (CNN) for the identification of spatial attributes in images with LSB Steganography, from images coming from BOSSbase (Break Our Steganography System) v1.01, a database of grayscale images designed to execute steganalysis tests. This database contains 10,000 carrier images and images with steganography, with 50% distribution and spatial resolution of 512x512. 70% of this dataset was assigned to model training, 10% to validation and 20% to testing. Five densities of embedded content (payload) were considered: 0.1, 0.2, 0.3, 0.4 and 0.5 bpp (bits per pixel). The classifier achieved a performance measured in accuracy of between 84% and 86%.

Following the implementation of CNN for detecting steganography in monochromatic images, Wu et al. [34] developed in 2016 a learning model with deep residual networks (DRN) from 10,000 images with spatial resolution 512x512, coming from the database called BOSSbase (Break Our Steganography System) v1.01. The authors extracted four non-overlapping windows of 256x256 from each image, in a way that a set of 40,000 input images were settled, with a density of embedded content (payload) of 0.4 bpp. They used five known algorithms to embed steganographic content to input images: HUGO-BD (Undetectable steganography With Bounding Distortion), WOW (Wavelet Obtained Weights steganography), HILL (High-pass Low-pass Low-pass steganography),

Table 1 State of the art relevant to this work of algorithms for the detection of steganography in images

Method	Ref	Year	Input data base	Steganography algorithm	Payload (bpp)	Spatial resolution	Number of observations	Accuracy (%)
Artificial Neural Networks	[14]	2016	Non reported	steganPEG and quickstego	Not reported	128x128 to 512x512	Train (75%): 150 Test (25%): 50 Total: 200	97 to 99
Artificial Neural Networks	[2]	2017	Non reported	Non reported	0.1, 0.25	512x512	Train (70%): 3,360 Test (30%): 1,440 Total: 4,800	86 to 90
Convolutional Neural Networks	[26]	2016	BOSSbase v1.01	Proposed by the authors	[0,1, 0.5]	512x512	Train (80%): 8,000 Test (20%): 2,000 Total: 10,000	84 to 86
Convolutional Neural Networks	[34]	2016	BOSSbase v1.01	HUGO, WOW, S-HILL UNIWARD and MPOD	0.4	256x256	Train (50%): 20,000 Test (50%): 20,000 Total: 40,000	89 to 96
Convolutional Neural Networks	[35]	2017	BOSSbase v1.01	HUGO-BD, DC, and S-HILL UNIWARD	0.05, 0.1, 0.2, 0.3 and 0.4	512x512	Train (50%): 5,000 Test (50%): 5,000 Total: 10,000	68 to 84
Convolutional Neural Networks	[30]	2017	BOSSbase v1.01	HUGO and S-HILL UNIWARD	0.1 and 0.4	512x512	Train (80%): 8,000 Test (20%): 2,000 Total: 10,000	70
Convolutional Neural Networks	[17]	2017	BOSSbase v1.01 and SPHP	Proposed by the authors	0.4	512x512	Train (75%): 30,000 Test (25%): 10,000 Total: 40,000	96 to 98
Convolutional Neural Networks	[4]	2017	Non reported	Proposed by the authors	0.1, 0.25, 0.5 and 1.0	512x512	Train (60%): 6,000 Test (40%): 4,000 Total: 10,000	95 to 96
Convolutional Neural Networks	[27]	2017	BOSSbase v1.01 and SPHP	Proposed by the authors	0.5	256x256	Train (80%): 8,000 Test (20%): 2,000 Total: 10,000	72 to 88
Convolutional Neural Networks	[32]	2017	BOSSbase v1.01 and BOWS2	Proposed by the authors	0.1 and 0.5	512x512	Train (70%): 8,120 Test (30%): 3,480 Total: 11,600	91 to 92

S-UNIWARD and MiPOD. 50% of this data set was assigned for model training and 50% for testing. The classifier's performance measured in accuracy ranked between 89% and 96%.

In 2017, Wu et al. [35] used the same learning model based on CNN proposed in 2016 for image binary classification with LSB steganography, starting from images coming from the BOSSbase database (Break Our Steganography System) v1.01. They used again four known algorithms for embedding steganographic content to input images: HUGO-BD, DC, and S-UNIWARD HILLF. Five densities of embedded content (payload) were considered for this study: 0.05, 0.1, 0.2, 0.3 and 0.4 bpp (bits per pixel). 50% of this data set was assigned for model training and 50% for testing. The classifier achieved an accuracy between 68% and 84%.

Likewise, Sharifzadeh et al. [30] developed in 2017 a learning model based on CNN for binary classification of images with LSB steganography, from 10,000 images with a spatial resolution of 512x512 coming from BOSSbase database (Break Our Steganography System) v1.01. The authors used three known algorithms for embedding steganographic content to input images: HUGO, HILL, and S-UNIWARD. Two densities of embedded content (payload) were considered: 0.1 and 0.4 bpp, evenly distributed over the set of input images. 80% of the pairs of images (carrier and steganography) was assigned for training model and 20% for testing. The classifier achieved an accuracy of about 70%.

In 2017, Kim and Lee [17, 18] developed a system based on CNN for the identification of images with steganography in the spatial domain, considering two preprocessing stages: filtering by means of a high pass filter (HPF) and filtering through a binarization differential filter (BDF). The authors used 10,000 images for the study, with a spatial resolution of 512x512, coming from BOSSbase (Break Our Steganography System) database v1.01 and SIPI. Four non-overlapping windows of 256x256 from each image were extracted by the authors, and a steganography algorithm with two configurations was implemented, in such a way that a set of 80,000 input image was formed. The density of embedded content (payload) was 0.4 bpp. 75% of this dataset was assigned for the model training and 25% for testing. The classifier's accuracy was between 96% and 98%.

Chhikara and Kumari [4] developed in 2017 a binary classification model for detecting steganography on still images, focusing on the feature selection process. This process was based on the Wrapper GLBPSO method, which allowed the authors the detection of the interactions among variables, considering the Discrete Cosine Transformation (DCT) [25] and the Subtraction Pixel Adjacency model (SPAM) [24]. The data set consisted of 5,000 pairs of JPEG images distributed in 60% for training and 40% for testing. The steganographic content was embedded considering four densities of embedded content (payload): 10%, 25%, 50% and 100% of the total size of the input image. Each image was cropped in windows with a spatial resolution of 8 rows and 8 columns. The features were extracted from these windows, which served as the input of the input neurons of the binary classification model. The classifier achieved an accuracy of between 95% and 96%.

In 2017, Aljarf et al. [2] developed a system based on artificial neural network and multilayer perceptrons (MLP) for binary classification of grayscale images and color images with LSB steganography, based on image histogram features. The authors identified that the peak value of the histogram decreased when LSB content was embedded, while the re-normalized histogram (the radius of the histogram to the peak value) increased. The set of 4,800 input images used by the authors was distributed as follows: 2,400 grayscale images (1,200 carriers and 1,200 with steganography) and 2,400 color images (1,200 carriers and 1,200 with steganography). All images counted on a spatial resolution of 512x512 and two densities of embedded content (payload) were considered: 10% and 25% of the total size of

the input image. 70% of pairs of images (carriers and with steganography) were assigned for the model training and 30% for testing. The classifier achieved a performance measured in accuracy between 86% and 90%.

A precise focus on the implementation of standardized algorithms of steganalysis was executed by Qian et al. [27] in 2018, who developed a CNN model to assess the learning attributes of steganalysis by applying the HUGO, WOW, S-UNIWARD, MiPOD, and HILL algorithms on a set of 5,000 images with spatial resolution of 256x256. These images were adapted from the BOSSbase (Break Our Steganography System) database v1.01, forming 5,000 pairs of input images (carriers and with steganography). The images were transformed by running reflections on each of the axes, with which were formed 100,000 images in total. A density of embedded content (payload) of 0.5 bpp was considered. 80% of pairs of images (carriers and with steganography) was assigned for model training and 20% for testing. The classifier achieved an accuracy between 72% and 88%.

In 2019, Sun et al. [32] developed a learning model based on CNN for binary classification of images with LSB steganography, from 10,000 images with a spatial resolution of 512x512 from BOSSbase (Break Our Steganography System) databases v1.01 and BOWS2. The authors extracted four windows from each image, in a way that an initial set of 40,000 input images with a density of embedded content of 0.1 bpp was formed. From this set were extracted 5,800 pairs of images. 70% of the pairs of images (carriers and with steganography) was allocated for model training. 30% of the set of input images was divided into 10 sets of 50, 20 and 10 images each one, and three sets were chosen to execute the tests. A fourth set was formed with the same images of one of the three sets chosen, but with a variation in the density of embedded content, setting it in 0.5 bpp. The classifier achieved a performance measured in accuracy between 91% and 92%.

3 Methodology

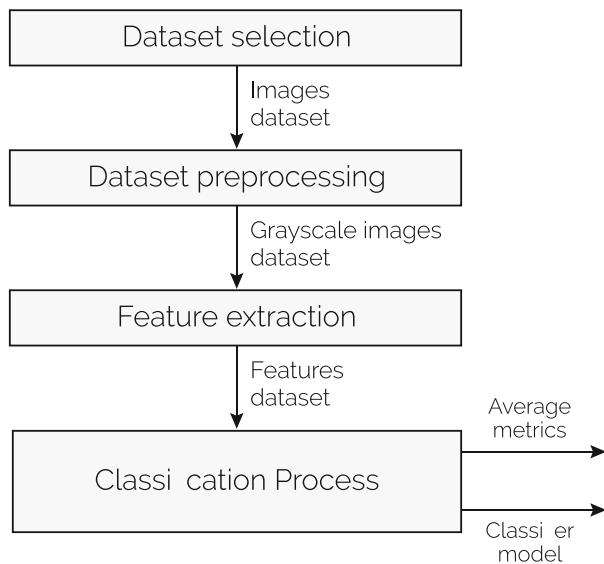
The methodology followed in this work is shown in Fig. 1. First, a selection of the dataset was performed. Subsequently, a preprocessing of the images was made in order to obtain a set of pairs of monochromatic images with a uniform spatial resolution with and without LSB steganography. These pairs of images were entered into the classification process that identified the existence of steganography. Here is explained in more detail each of these stages.

3.1 Dataset selection

The set of sample images was taken from Standford's database derived from the Genome Project, available in [19]. This database contains an image population of 108,077 RGB images in JPEG format, labeled and categorized in 80,138 different categories, with spatial resolutions from 72 pixels in length and width, up to 1,280 pixels width and length, with an average of 500 pixels width and length. This allows us to state that the database is heterogeneous as for kinds and spatial resolutions, which makes the task of classification by means of neural network models head to be generalizable.

From this population was extracted an initial sample of color images, considering two criteria: the homogenization of the spatial resolution and the temporary complexity in the preprocessing and classification model training, as documented [35] and [27].

Fig. 1 Block diagram of the methodology proposed for the detection of LSB steganography

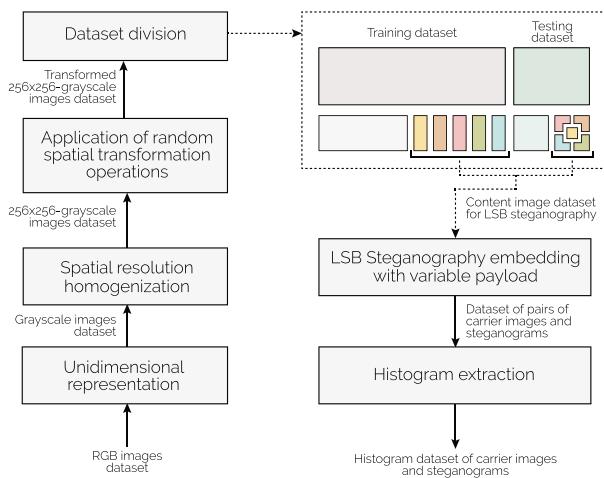


3.2 Preprocessing of the image dataset

The preprocessing of the set of selected images was performed by using the Matlab work environment in its R2018a version with educational license number 160127 and Data Acquisition Toolbox and Communications Toolbox.

The whole preprocessing process is shown in Fig. 2. First, we proceed to represent color images chosen in the previous step in a one-dimensional color model. Thus, 70.000 JPEG monochromatic images with a fixed spatial resolution of 256x256 and a depth of 8 bits (256 levels of intensity from 0 to 255) are obtained. Subsequently, the set is divided into two datasets, one for training (80% of monochromatic images chosen randomly) and one for testing (20% of monochromatic images chosen randomly). Each of these two sets was halved to form pairs of carrier images and images with LSB steganography so that the

Fig. 2 Block diagram of the data-set preprocessing stage



binary classification model would be trained with an equal number of observations, which avoids an imbalance of classes.

These division processes were carried out by choosing images randomly from each of the sets established. Finally, the set of monochromatic images that were going to be altered with steganography was divided randomly and equally into five sub-kinds which identify the five loads of density of steganographic content (payloads): 0.1, 0.2, 0.3, 0.4 and 0.5. Each of these subclasses had a total of 5,600 observations. While each set was sectioned, a function of random transformation on monochrome images comprising seven transformations was applied:

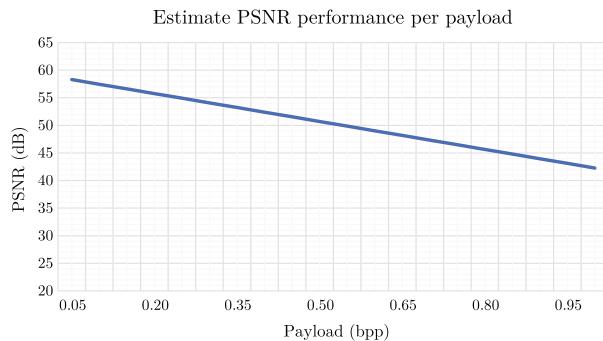
- 0° clockwise image rotation (r_{0°).
- 90° clockwise image rotation (r_{90°).
- 180° clockwise image rotation (r_{180°).
- 270° clockwise image rotation (r_{270°).
- Image reflection around the vertical axis (R_v).
- Image reflection around the horizontal axis (R_h).
- Transpose of the image matrix (T_{image}).

The application of these transformations was made once for each image randomly and the carrier images were stored in the BMP format to prevent image compression which may affect the spatial variations executed.

An LSB steganography embedding algorithm used to hide graphic content in monochrome images is based on the algorithm presented by [6]. This algorithm is a variation of the traditional LSB steganography algorithm, with a constant message image. A random sequence based on a Pseudo-Random Number Generators (PRNG) is defined, and with the sequence, the pixels to be modified are assigned. Additionally, an XOR operation is made between the least significant bit of the pixels to be assigned of the carrier image, with the message's bits. This procedure makes it difficult to detect the message image, as documented by [6]. The mean Peak Signal-to-Noise Ratio (PSNR) of our algorithm is 54.08 dB (51.24 dB in the worst case with payload 0.5 bpp and standard deviation of 0.027 dB, and 58.01 dB in the best case with payload 0.1 bpp and standard deviation of 0.085 dB), which outperforms the initial results described by [6].

An estimate of the general performance of the steganographic algorithm measured by PSNR is shown in Fig. 3, in which it is evident that when increasing the payload, the PSNR decreases linearly with a slope of 1.70 dB for each 0.1 bpp increase in the payload. The estimation has an R-squared of 0.96.

Fig. 3 Performance of the proposed steganographic algorithm measured with PSNR, when the payload is increased



3.3 Feature selection and extraction

Once the dataset of pairs of carrier images and steganograms was processed, the following descriptors were extracted, which represented each of the input images¹: (a) Histogram kurtosis, (b) Histogram skewness, (c) Histogram standard deviation, (d) Histogram range, (e) Histogram median, (f) Histogram geometric median, (g) Histogram Hjorth mobility, and (h) Histogram Hjorth complexity. The first six features allow to quantify small graphical variations not visually perceptible in the images, while the last two attributes allow to give a non-linear description of these variations.

Other features that were considered but not included in the training, validation, and testing of the neural network were: (a) Interquartile ranges (0.1, 0.3 and 0.9) of the histogram, (b) Histogram harmonic mean, (c) Histogram correlation dimension, (d) Histogram maximum Lyapunov exponent, and (e) Histogram Hurst Exponent.

The first three features were dismissed because they do not describe variations in the dynamics of the embedded of LSB steganographic content, unlike the statistical attributes chosen. The last two attributes were discarded because of the computational cost, which implied to calculate them since polynomial adjustments must be made continuously until there was convergence, which required heavy use of resources such as time, memory and processing.

3.4 Classification process

Once processed the set of images, the process of binary classification of 256x256 monochromatic still images run in two classes: a negative class corresponding to carrier images and a positive class corresponding to steganogram images with LSB steganography. For the detailed process of binary classification were proposed six models based on artificial neural networks, five models for each individual and independent payloads, and one general classification model with a centralized function that grouped the observations of all payloads interchangeably.

Models of artificial neural networks were proposed because:

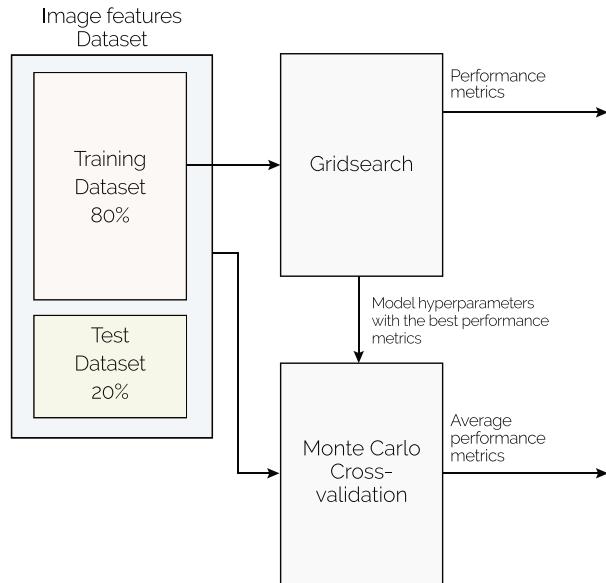
1. They have a better general performance in the multiclass classification, and are aligned with the objective that is being proposed within this research framework.
2. When the amount of training data is large, the ANN models outperform SVM models and logistic regression, among others.
3. ANN models can be trained in one go, a characteristic required to reduce the time complexity of training.
4. It was necessary to stipulate a fixed architecture with which the training and testing could be executed in such a way that the number of observations did not influence its structure. This situation was remedied with a parametric ANN approach.

The overall classification process is shown in Figs. 4, 5, and 6, comprising the following procedure:

1. The hyperparameter to be optimized by means of Grid search and performance metrics is defined (Fig. 5).

¹ Public dataset at: <http://dx.doi.org/10.21227/gs67-yn65>

Fig. 4 Overall classification process



2. N iterations are executed, corresponding to the number of combinations of hyperparameters. In each iteration, K-Fold cross-validation with 5 folds was used to determine the metrics of each model for each combination of hyperparameters.
3. The hyperparameter combination with better metrics was chosen and this neural network model was used in the validation of Monte Carlo.
4. Five (m value) iterations were executed, in which the chosen model was trained through grid search, with the dataset distributed five times randomly between training set (64%), validation set (16%) and testing set (20 %).
5. The metrics of the performance model were calculated by averaging the metrics provided by the iterations, which were obtained as a result of the validation of Monte Carlo, a process explained in the section of performance evaluation (Fig. 6).

For the Grid Search process, optimization algorithms were considered: Adadelta, RMSprop, and Adam, as proposed by [29], who concludes that such optimization algorithms have a similar performance in similar situations. As for the size batch and the number of epochs, the ranges of values [8, 16, 32, 64, 128] and [10, 50, 100, 250, 500] were considered respectively. These ranges were raised according to the results shown by [33]

Fig. 5 Optimization of hyperparameters by grid search

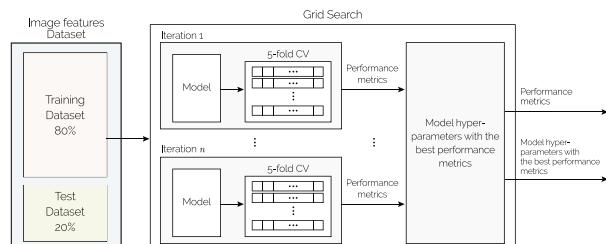
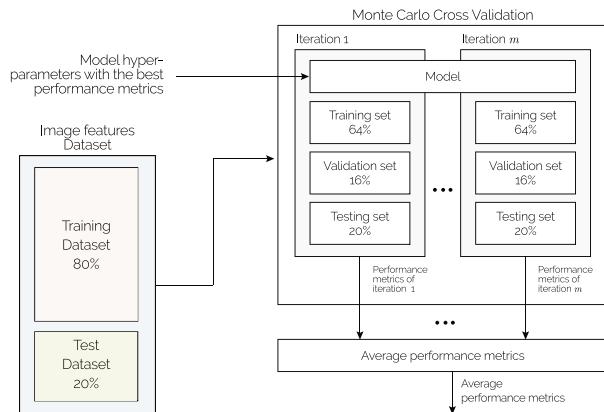


Fig. 6 Monte Carlo cross-validation



wherein the performance of neural networks, the batch size and the number of epochs in different ranges, applied to known datasets.

3.5 Performance evaluation

The metric used to evaluate the performance of each of the classification models proposed in the hyperparameter tuning by grid search was accuracy, as with this metric it was possible to generalize the model behavior and also compare its performance against the performance obtained by other authors related to this context. The performance metric was calculated from the relationship between true positives, true negatives, false positives and false negatives of each of the confusion matrixes obtained in the validation process and the hyperparameter tuning.

To increase the reliability of the metrics associated with the classification model, Monte Carlo cross-validation (Fig. 6) was executed. In this method, based on an input neural network model, m training iterations are carried out and m sets of metrics are calculated by distributing the observations of the whole dataset into groups of the training set, validation set, and testing set, as stated in the previous section. The final average metric calculated was accuracy.

Accuracy was the metric calculated to measure the performance of the model. This metric allows you to evaluate binary classifiers by the proportion of correct predictions over the total number of evaluated observations:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

In Eq. (1), TP refers to true positives (correctly classified steganography observations), TN to true negatives (correctly classified non-steganography observations), FP to false positives (incorrectly classified steganography observations), and FN to false negatives (misclassified non-steganography observations). Accuracy was the selected metric considering that (a) the dataset presents a balance between the classes in which 50% belong to the positive class and 50% belong to the negative class, (b) we have not defined a weight on the importance of a false positive or a false negative, and (c) accuracy allows us to compare the results with respect to related studies.

Even though we calculate the accuracy to present the performance of the model, other additional metrics are reported to make the study more comparable:

- Precision, also called positive predictive value, measures the proportion between the relevant (positive) observations over the total positive observations:

$$\text{precision} = \frac{TP}{TP + FP} \quad (2)$$

- Specificity, also called true negative rate, measures the proportion of observations classified as negative compared to the total number of negative observations:

$$\text{specificity} = \frac{TN}{TN + FP} \quad (3)$$

- Sensitivity, also called the true positive rate or Recall, measures the proportion of observations classified as positive compared to the total of positive observations:

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (4)$$

- F1 Score, corresponds to the harmonic mean between the precision and the recall:

$$F1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

In addition, the classification model was tested individually against a dataset of pairs of carrier images and steganograms for each stipulated payload (0.1, 0.2, 0.3, 0.4 and 0.5), in order to generalize the overall performance of the classification model, as well as to observe the incidence of each payload individually. The results are presented in the following section.

4 Results

The following section presents the results obtained by applying the methodological processes detailed in the former section, following the same order presented in that section.

4.1 Dataset selection

From the population of 108.077 color images in JPEG format, labeled and categorized an initial sample of color images was extracted, considering the two criteria established in the former section. By analyzing these criteria, 70.000 color JPEG images with a fixed spatial resolution of 256x256 and a bit depth of 24 were selected.

4.2 Pre-processing of the image dataset

Considering the color model of the images in the selected sample, when representing the images as one-dimensional arrays, the images following the intensity equation below Eq. (6) are obtained [22]:

$$Gray = 0.299 * R + 0.587 * G + 0.114 * B \quad (6)$$

Let $Gray$ be the value of one-dimensional intensity and the triplet (R, G, B) , the intensity values of the three components of color images. Afterward, spatial transformation operations were applied randomly to the input images, obtaining what is shown in Fig. 7, where the seven transformations applied can be observed.

It should be considered that the maximum content to embed is one-eighth of the spatial resolution of the carrier image, that is, an image of a maximum spatial resolution of 90x90. The relationship between the number of pixels to be modified in the carrier image ($SR_{carrier}$) with respect to the spatial resolution of the carrier image (width $W_{carrier}$ and height $H_{carrier}$) and the payload (P) is the following Eq. (7):

$$SR_{carrier} = \frac{W_{carrier} * H_{carrier}}{8} * P \quad (7)$$

Thus, for the five density loads, the spatial resolution of the graphic content, considering a square matrix of representation with respect to the fixed resolution of the carrier image of 256x256, will be the square root of the number of pixels modified, as shown in Table 2.

Fig. 7 Example of application of spatial transformation operations. From left-to-right and top-to-bottom: (a) 0° clockwise rotation (original image), (b) 90° clockwise rotation, (c) 180° clockwise rotation, (d) 270° clockwise rotation, (e) reflection around the vertical axis, (f) reflection around the horizontal axis, and (g) image transpose

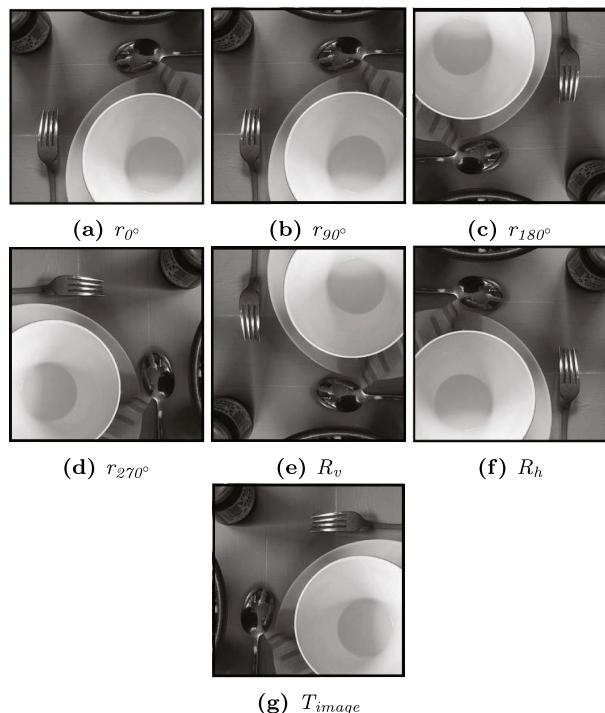


Table 2 Relationship between the density load of the message (payload) embedded and the spatial resolution message

Payload	Spatial resolution of graphic content embedded
0.1	28x28
0.2	40x40
0.3	49x49
0.4	57x57
0.5	64x64
1.0	90x90

In this way, the dataset of monochromatic images is subdivided into subsets of images, as shown in Fig. 8.

Figure 9 shows an example of the result of executing the preprocessing stage and the execution of the steganography algorithm on a carrier image (Fig. 9a), embedding a content image of spatial resolution 64x64 and payload 0.1, which produces a graphic steganogram (Fig. 9b).

The steganogram images were stored in BMP format to avoid image compression affecting spatial variations executed by means of the LSB steganography algorithm.

Fig. 8 Diagram of observations distribution that served as input to the classification model

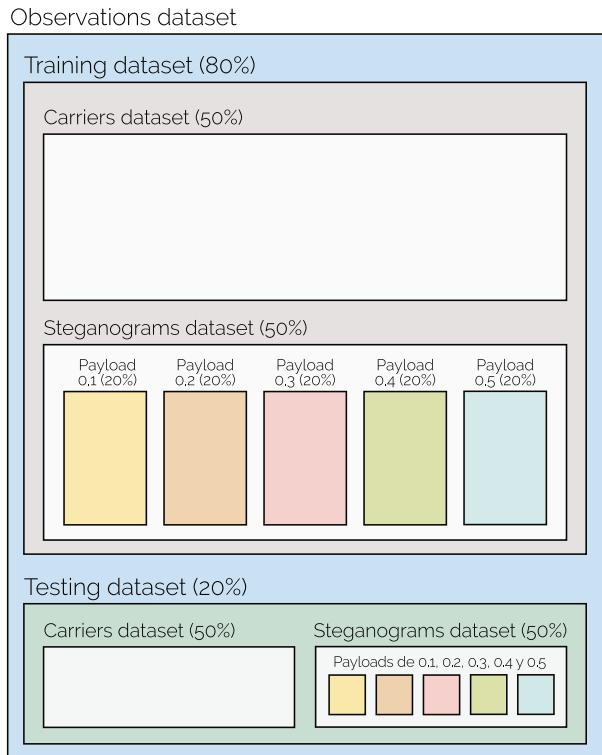
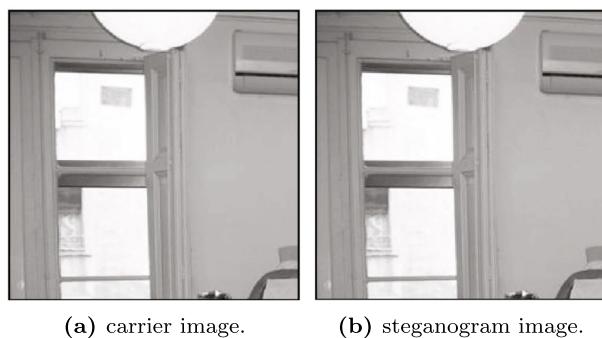


Fig. 9 Example of application of the steganography algorithm during the preprocessing stage with a payload of 0.1: (a) the carrier image, and (b) the steganogram



4.3 Feature selection and extraction

The features were calculated from the histogram of each carrier pair and steganogram. An example of the histogram of a dataset pair is shown in Fig. 10.

As can be seen, there are slight differences between the two histograms, which are visible to the naked eye, such as the frequency of some intensity levels and the distribution of intensities along with the histogram. When extracting features from the pairs of carrier images and processed steganogram, was obtained what is shown in Fig. 11, which shows the behavior of the mobility attribute, one of the eight attributes chosen as input to the neural network model.

In the graph, the horizontal axis represents the ID of each of the test images and the vertical axis shows the variation in the magnitude of Hjorth mobility, calculated from the histogram of each image. The blue curve shows the behavior of the mobility feature for the carrier images and the orange curve shows the behavior for the steganogram images. The steganograms images from 0 to 399 have a payload of 0.1; from 400 to 799 they have a payload of 0.2; from 800 to 1199 they have a payload of 0.3; from 1200 to 1599 they have a payload of 0.4; from 1600 to 1999 they have a payload of 0.5. It may be noted that there is a very strong dynamics in the mobility that is probably related to the variation in the payload of the steganogram images, which allows to state at first sight that this feature can successfully identify the existence of steganography through the analysis of its dynamics. The other seven attributes that were chosen also showed heterogeneous dynamics that were related to the inclusion and change of steganographic content in the carrier images.

Fig. 10 Carrier and steganogram histograms

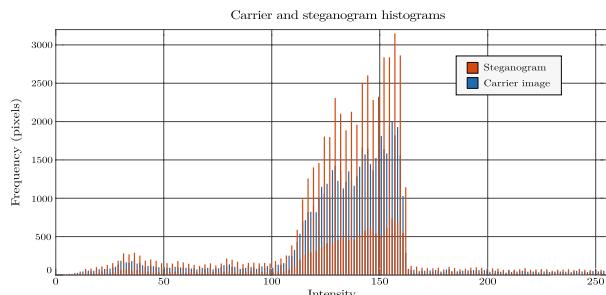
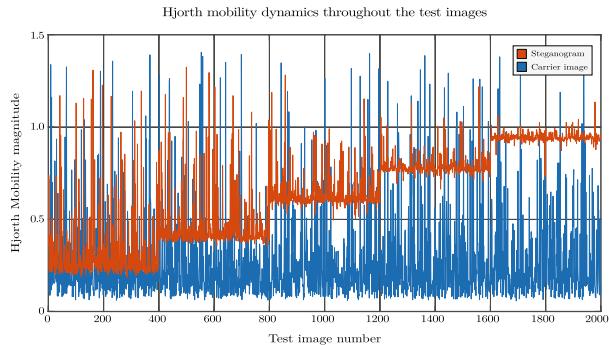


Fig. 11 The behavior of the Hjorth mobility attribute in the carrier images (blue curve) and steganograms (orange curve) as the payload is increased



4.4 Classification process

The characteristics of the architecture proposed for the classifier were established in accordance with the hyperparameters resulting from the application of the optimization process by means of Grid Search and considering the criteria proposed by [1, 11–13, 15, 21]. The following table presents the architecture proposed for the best ANN model used for the classification process (see Table 3), considering the 10-fold Cross-Validation:

The neural network model has an architecture in which there are eight input neurons in the input layer (one neuron per input hyperparameter), two hidden layers and one output neuron, since it is a binary classification. A batch size of 64 was used, which represents the number of observations per training cycle, and a total of 250 complete training cycles through the training data (epochs).

4.5 Performance evaluation

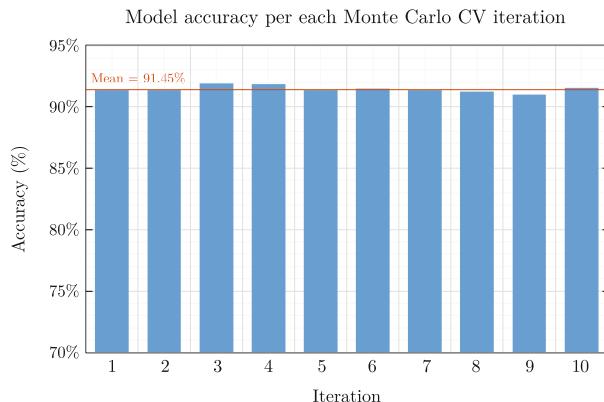
When executing ten training iterations and calculating the metric sets of Monte Carlo cross-validation, the results shown in Fig. 12 are obtained, where the horizontal axis represents the number of iterations and the vertical axis shows the accuracy per iteration, and the constant orange line represents the average accuracy (91.45%) of the ten iterations shown.

It may be noted that a low variance of 0.068% is present, which indicates that the accuracy is considerably homogeneous throughout the execution of the test iterations of the model, meaning that it is highly generalizable for the context of LSB steganography

Table 3 The binary classification model architecture for LSB steganography detection

Characteristic	Value
Number of hidden layers	2
Number of neurons per concealed layer	4 and 2
Number of input neurons	8
Number of output neurons	One
Activation function of the input layer	Rectifier
Activation function of concealed layers	Rectifier
Activation function of the output layer	Sigmoid
Optimizer	Adam
Batch size	64

Fig. 12 Accuracy of the classification model along with the iterations of the Monte Carlo cross-validation



detection. The accuracy closeness of the iterations with the average value indicates a low variance.

Some additional average metrics of the best performance model, after having applied Monte Carlo cross-validation are the following (see Table 4):

When matching the classification model to 2,800 pairs of individual pairs of observations (carriers and steganograms) for each payload defined, the classifier's performance per such variations is the one shown in Fig. 13, where the horizontal axis represents the payload and the vertical axis represents the accuracy of the model when classifying the observations of each of the payloads.

It can be seen in the graph that there is a considerable difference in the classifier's performance as variations in the density of the content message in the steganogram are executed. For messages with a low performance inside the carrier image, the model turns out to be less accurate before its detection. For messages with greater embedded content in a carrier of the same size, the classifier's accuracy increases.

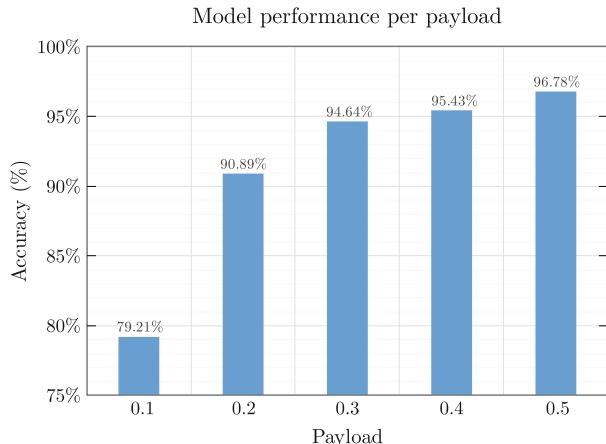
5 Discussion

LSB steganography detection turns out to be particularly complex when it comes to graphic media such as still images, as the modification of the least significant bits does not have repercussions on a visually appreciable variation. The significance of this study lies (a) in the detection of images with LSB steganography of variable payload, from 0.1 to 0.5 bpp, (b) the application of the classifier to detect individual payloads, (c) the construction of a dataset which balances the categories and classes, and (d) the application of transformation operations to the input data in order to ensure that the model does not learn spatial

Table 4 Average metrics resulting from the execution of the Monte Carlo cross-validation process with the best binary classification model

Metrics	Average value (%)
Sensitivity	88.97
Specificity	93.93
Precision	93.77
F1-Score	91.22

Fig. 13 Accuracy variation when the message payload is increased in the steganogram



patterns and that there are no horizontal or vertical trends which affect the classification of the embedded content.

Representing images in a one-dimensional way abbreviates computational processes in time and space, as well as the identification of embedded content by means of descriptors calculated from its histogram. This allows to evidence slight variations in the distribution of their intensity and frequency, which can be taken advantage of by the descriptors in order to accentuate the behavior of the LSB steganography vector used to conceal the message contained in the carrier images.

The sensitivity and accuracy metrics that measure the classifier's performance increased as the steganography load of the message also increased on the carriers. This may be due to intensity modifications on the carrier as the payload is increasing. They are more recurrent, which makes more pixels with even intensities increase by one unit, and pixels with odd intensities decrement by a unit. Additionally, as shown in the feature selection, their behavior is heterogeneous before the payload variation, which makes the dynamics get influenced by the amount of embedded content. Conversely, since the message is not homogeneous and does not present a distinctive pattern in its intensities, indistinct variations in the carrier's pixels help to recognize the steganogram.

In complement, accuracy and sensitivity are metrics in which false negatives are considered as a preponderant variable, which means that an increase in these means a direct decrease of accuracy and sensitivity. It is for this reason that when increasing the content of steganography, the neural network model of the classifier has the capacity to detect the steganograms with better efficiency, which results in a decrease in false negatives, increasing the metrics outlined.

On the other hand, the results obtained showed 91.45% overall accuracy and 96.78% by individual maximum accuracy per payload in the binary classification of images with varying LSB steganography generated from a population of images from the Genome Project's database. These results are consistent with those reported by other authors: Qian et al. who obtained 90% accuracy in the classification in 2016 with payloads from 0.1 to 0.5, Wu et al. who obtained 86% accuracy in the classification in 2016 with payloads from 0.05 to 0.5, Chhikara and Kumari who obtained 96% in the best case of the classification approach in 2017 with payloads from 0.1 to 1.0, and Sun et al. who obtained 92% accuracy in the classification in 2019 with payloads of 0.1 and 0.5. With the model presented in this

article, we have achieved an accuracy equivalent to that presented by the authors, detecting a wide range of payloads, which is a significant contribution. Additionally, in the best of cases, the presented classifier outperforms those studied and presented in Table 1, which is promising for future work.

6 Conclusions

The model for LSB steganography detection proposed enabled the binary classification of carriers and steganograms by means of its pre-processing, feature extraction, training, validation, and testing processes. The narrow margin of difference between the performance metrics per iteration of the Monte Carlo cross-validation allowed to guarantee the generalization of the classification model.

Thanks to the one-dimensional representation processes, image cropping with fixed spatial resolutions of 256x256 and feature extraction, features of interest in steganograms were highlighted, and it was possible to associate the payload measurements with the descriptors chosen. Kurtosis, Asymmetry, Standard Deviation, Range, Median, Geometric Median, Hjorth Mobility and Hjorth Complexity were eight features considered; all calculated from image histograms. The latter two attributes allowed an improvement in the classifier's performance, as they provided a description of the non-linear dynamics of steganography in the observations.

According to the results presented, the following future research studies are proposed: developing a voting detection system, considering individual models for the detection of LSB steganography and considering a binary classification model per payload; studying the dependence between features for the reduction of temporary complexity; selecting additional features which allow the model to learn from the non-linear dynamics of steganography algorithms; and replicating the LSB steganography detection model to other contexts of this steganographic area for the detection of embedded images in other media objects.

Acknowledgements The Bolivarian Pontifical University of Bucaramanga supported this research. Thanks to Social Communicator and Journalist Daniela Gomez-Diaz, who constantly accompanied the algorithmic development of the model proposed and provided important support throughout the research.

Funding This study was funded by Pontifical Bolivarian University (in kind money).

Declarations

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest Author A declares that he/she has no conflict of interest. Author B declares that he/she has no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Agarp A (2019) Deep learning rectified using linear units (Relu). Department of Computer Science Adamson University, p 7
2. Aljarf A, Amin S, Shuttelworth J, Filippas J (2017) Detection system for gray and color images based on extracting features of difference image and renormalized histogram. *Journal of Information Hiding and Multimedia Signal Processing* 8(2):16
3. Chandrababu A (2009) Using an artificial neural network to detect the presence of image steganography. The Graduate Faculty of The University of Akron, Akron, Ohio, United States
4. Chhikara R, Kumari M (2017) Blind image steganalysis using neural networks and wrapper feature selection. 2017 International Conference on Computing, Communication and Automation (ICCCA). Greater Noida, India
5. Cox J (2018) Shakespeare, tiny picture this on twitter contains the complete works of, Vice Media LLC. [Online]. Available at: <https://bit.ly/2WZJQYZ>. Accessed 25 Nov 2018
6. Emam MM, Aly AA, Omara FA (2016) An improved image steganography method based on LSB technique with random pixel selection. (*IJACSA*) Int J Adv Comput Sci Appl 6
7. Fairchild M (2013) Color appearance models, 2nd edn. Munsell Color Science Laboratory, Rochester Institute of Technology, John Wiley & Sons Ltd, p 1-34
8. Fridrich J, Soukal D, Goljan M (2005) Maximum likelihood estimation of the length of secret message embedded using steganography + k in spatial domain. *Security, Steganography, and Watermarking of Multimedia Contents VII*. San Jose, California
9. GeoEdge PR Department (2018) Embedded malicious ads in Ad images according to gain traction GeoEdge's research, Geoedge. [Online]. Available at: <https://bit.ly/2XxR5vx>. Accessed 25 Nov 2018
10. Gonzalez R, Woods R (2009) Digital image processing using MATLAB, 2nd edn. McGraw-Hill, p 826
11. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. Chapter 6: Deep Feedforward Networks, Boston: MIT Press, pp 168–224
12. Hansson M, Olsson C (2017) Feedforward neural networks Relu with activation functions are linear splines
13. Huang G-B (2003) Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Trans Neural Netw* 14(2):274–281
14. Ingale AK, Dharwadkar NV, Kodulkar P (2016) Universal steganalysis using DWT and entropy. 2016 International Conference on Signal and Information Processing (IConSIP). Vishnupuri, India
15. Janocha K, Czarnecki WM (2017) On loss functions for deep neural networks in classification. *Theoretical Foundations of Machine Learning 2017 (TFML 2017)*, p 10
16. Judd D (1952) Color in business, science and industry, 3rd edn. WS i. P. a. A. Optics, Ed, New York: Wiley Interscience, p 388
17. Kim D-H, Lee H-Y (2017) Convolutional neural network-based steganalysis on spatial domain. *Int J Math Comput Simul* 11:5
18. Kim D-H, Lee H-Y (2017) Deep learning-based steganalysis against spatial domain steganography. Bern, Switzerland
19. Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li L-J, Shamma D, Bernstein M, Fei-Fei L (2016) Visual genome: connecting language and vision using crowdsourced dense image annotations. California, USA
20. Kolata G (2001) Veiled messages of terror may lurk in cyberspace. *The New York Times Magazine*, United States, p 4
21. Larochelle H, Bengio Y, Louradour J, Lamblin P (2009) Exploring strategies for training deep neural networks. *J Mach Learn Res* 10:40
22. Mathworks' Algoritmos Machine Learning for classification (SVM) (2018) Mathworks. [Online]. Available at: <https://es.mathworks.com/discovery/support-vector-machine.html>. Accessed 22 May 2018
23. Patestas M, Gartner L (2006) A textbook of neuroanatomy. Blackwell Publishing, pp 256–259
24. Pevny T, Bas P, Fridrich J (2010) Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans Inf Forensics Secur* 5(2):215–224
25. Pevny T, Fridrich J (2007) Merging Markov and DCT features for multi-class JPEG. *Proceedings of SPIE - The International Society for Optical Engineering*, p 13
26. Qian Y, Dong J, Wang W, Tan T (2016) Learning and representations for image steganalysis transferring using convolutional neural network, IEEE 2016 International Conference on Image Processing (ICIP). Phoenix, AZ, USA
27. Qian Y, Dong J, Wang W, Tan T (2018) Feature learning for steganalysis using convolutional neural networks. *Multimed Tools Appl* 77(15):19633–19657

28. Qin J, Xiang X, Wang MX (2010) A review on detection of LSB steganography matching. *J Inform Technol* 9(8):1725–1738
29. Ruder S (2017) An overview of gradient descent optimization algorithms. Cornell University Library, Ithaca, New York, p 14
30. Sharifzadeh M, Agarwal C, Aloraini M, Schonfeld D (2017) Convolutional neural network application to steganalysis's steganography, visual communications and image processing conference
31. Sterling B (2001) The year in ideas: A to Z: goes digital steganography. *The New York Times Magazine*, p 109
32. Sun Y, Zhang H, Zhang T, Wang R (2019) Deep neural networks for efficient steganographic payload location. *J Real-Time Image Proc* 16(3):635–647
33. Thoma M (2017) Analysis and optimization of convolutional neural network architectures. *arXiv*, p 134
34. Wu S, Zhong S-H, Liu Y (2016) Steganalysis via deep residual network, IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS). Wuhan, China
35. Wu S, Zhong S-H, Liu Y (2017) A novel convolutional neural network for image steganalysis with shared normalization. *IEEE Transactions on Multimedia*, p 12

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.