XXXVI Ibero-Latin American Congress on Computational Methods in Engineering
CILAMCE
Rio de Janeiro, 22-25 Nov 2015

PREDICTION OF STEGANOGRAPHY IN IMAGES USING A MATHEMATICAL MODEL

Natasha N. Paiva

Renato Portugal

{natasha, portugal}@lncc.br

National Laboratory for Scientific Computing - LNCC/MCTI

Av. Getulio Vargas, 333, 25651-075, Rio de Janeiro, Petropolis, Brazil

Pedro Carlos S. Lara

pedro.lara@cefet-rj.br

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – CEFET/RJ, Campus Petrópolis

Abstract. *Nowadays, besides the growing popularity of steganography, there is a big interest from researchers in the development of strategies that easily detects the use of steganography with low computational cost, in order to increase speedup and, at the same time, to achieve better solutions. Steganography is the art of hiding a secret information that makes it difficult to detect such a message. This information is hidden inside another information of minor relevance, as an image, a video or a text among others, for that context. In the case of this study, we use the image as a vehicle. In contrast with what is proposed in the literature, this study presents a new matemathical model to predict if there is a hidden message in an image. For this purpose, we have developed a software, that analyzes and classifies a pixel according to his color and neighborhood through statistics proximity measures. For within the context of an image, it is expected that the color change is gradual within small regions. In addition, this study also presents a comparison of the results obtained using two statistics proximity measures: Arithmetic Mean and Variance.*

Keywords: *Steganography; Mathematical model; Steganography in images.*

1 INTRODUCTION

1.1 Relevance of the theme

In general, image processing and analysis have two interestingly correlated features: the high-computational demand and the relative simplicity to develop computational methods for it (Rosenfeld, 2013), (Chan and Shen, 2005). Hence, motivated by the constant increase in computational power, processing techniques have been largely studied and utilized for improving the quality of the solutions.

One immediate consequence of these aspects is, for example, the greater applicability of these techniques in increasingly larger and complex problems (Whitman and Mattord, 2011). In special when it is applied to information security and it is notorious that information security is an area with great potential for research and development.

Information security has traditionally been ensured with data encryption and authentication techniques (Francia et al., 2009). Through the years, different generic data encryption standards have been developed. Although these encryption standards provide a high level of data protection, they are not efficient in the encryption of multimedia contents due to the large volume of digital image/video data. In addition, researchers are starting to utilize information hiding techniques to enhance the security level of data encryption methodologies. Information hiding conceals not only the content of the secret message, but also its very existence.

1.2 Related works and goals

In the context defined above, an important inspired strategy is to analyze colors proximity (Fridrich and Long, 2000), (Westfeld and Pfitzmann, 2000). This approach consists in evaluating the number of pairs of neighbor colors. That means that two single colors are distinguished only by the least significant bit. *Raw Quick Pairs - RQP* was developed following this model, however it only provides an estimate of the message size and cannot be applied to grayscale images.

Another interesting method, based on statistical analysis of pairs values, has been proposed in (Westfeld and Pfitzmann, 2000). The method evaluates the pairs od values changed during the steganography and it is presumed reliable when the message's position is known.

Another strategy is based on the permutation of gray levels of the image (Fridrich et al., 2001). The stego-detection technique in this case is based on analyzing how the number of regular and singular groups of permutation changes with the increase in message length embedded in the least significant bit plane.

Given the complex scenario discussed, this work aims at demonstrating the discovery of relevant existing relationships among distinct factors, whether extrinsic or intrinsic, that may be crucial in the detection of steganography on images.

However, instead of relying on traditional techniques as tried before, we will employ a new paradigm, based on pixel classification according to its neighborhood, using statistical measures.

This paper is organized as follows. In Section 2 we briefly describe the steganography technique. The particular mathematical model we created is presented in Section 3 and its implementation is shown in Section 4. In Section 5 we present the data set comprinsing records about images and explain how the experiments were conducted. In Section 6 we analyze the results obtained in order to evaluate the feasibility in effectively adopting statistical measures as a tool to aid in predicting information inside images. Finally, Section 7 wraps up the paper by discussing the results and summarizes what was presented.

2 STEGANOGRAPHY

Steganography is the art of hiding information. Its primary goal is that only the message recipient should know of its existence, as well as how to extract it, in contrast to the cryptography whose purpose is to make communication unintelligible to those who do not have the key. Steganography is the way of communicating secretly hiding an undercover message within other information not relevant to that context, so that makes it difficult to detect such a message. Video and sound files, digital images and other types of computer files containing irrelevant or redundant information can be used as vehicles for this art. The image used to receive steganography is called cover-image and after the message is hidden in it, is called stego-image.

2.1 Phenomenon Abstraction

We can define an image as a set of numbers which represent different light intensities. This numerical definition forms a grid and the individual points are defined as pixels, in other words, a matrix of finite points as shown in Fig. 1.

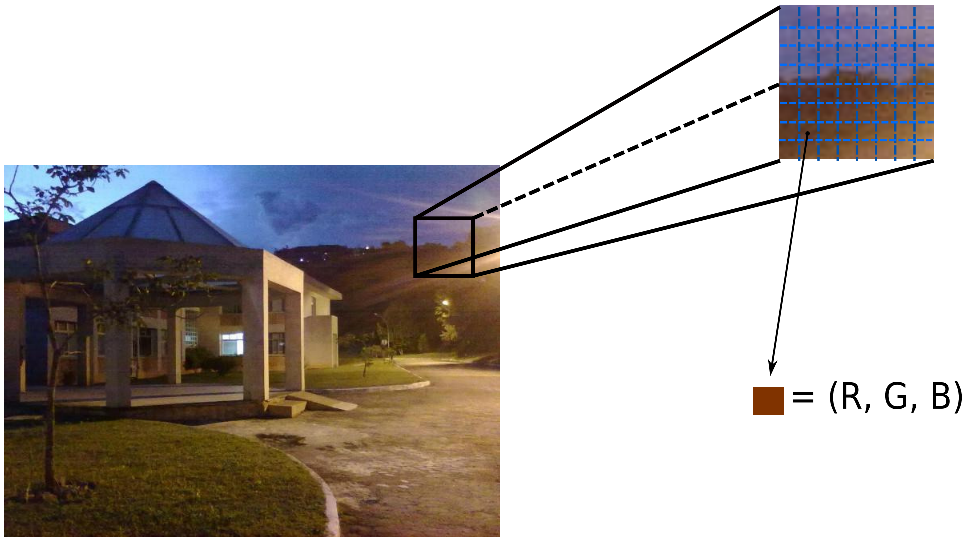


Figure 1: Physical composition of an image.

One of the possible ways to hide a message in an image is by changing the bits that composes it, in order to represent each letter in binary code, but in a way so subtle that is imperceptible. As shown in (Morkel et al., 2005), it changes the least significant bit of each byte.

Changing the Least Significant Bit - LSB, which is the last bit of each byte, is a simple insertion approach. The least significant bit of a few bytes is changed to form the message to be hidden. According to (Silman, 2001), when a 24-bit image is used, one bit for each color component can be used, since they are each represented by one byte. That is, you can store three bits in each pixel. For example, a representation of a 24-bit image can be seen below:

$$(11100110, 11110110, 11010010)$$
$$(11100110, 11110110, 11010001)$$
$$(11100101, 11110000, 11010010)$$

Suppose we want to hide the letter "o" in the image. This letter can be represented in binary as the following:

- "o" = (01101111).

Furthermore, we can assume that the pixels chosen to be modified are originally represented by the set of numbers:

$$(230, 246, 210), \quad (230, 246, 209), \quad (229, 240, 210)$$

or similarly, each value expressed in binary

$$(230, 246, 210) = (11100110, 11110110, 11010010)$$
$$(230, 246, 209) = (11100110, 11110110, 11010001)$$
$$(229, 240, 210) = (11100101, 11110000, 11010010).$$

To enter the letter "o" in these pixels, we modify the last digit of each element of each byte to represent the binary value of this letter, in order to finally obtain the modified pixels of the form:

$$(1110011\mathbf{0}, 1111011\mathbf{1}, 1101001\mathbf{1})$$
$$(1110011\mathbf{0}, 1111011\mathbf{1}, 1101000\mathbf{1})$$
$$(1110010\mathbf{1}, 1111000\mathbf{1}, 11010010)$$

Notice that the modified digit marked in bold form the binary representation of the letter and maintains the value of the byte that is very close to the original pixel value.

An approach that has become very effective is to convert the image into a matrix structure, allowing the classification of the pixels location in the modified image as correct. If we assume the target image as a matrix wich dimension depends on the number of pixels, where each entry of the matrix represents a pixel and would be composed of a set (R, G, B), being intensities of red, green and blue respectively, we can look for any patterns in the matrix associated with the target image according to the idea we have of an image without modification, in which the color variation is quite soft, indicating proximity between the values of a pixel and its immediate neighbors. With this idea, we can search pixels considered wrong in the target image.

3 MATHEMATICAL MODEL

In an attempt to develop a mathematical model that detects the presence of modifications in a target image, we can consider the image being composed of a total of $N \times M$ pixels. Then, we can associate the image to a matrix $\mathcal{P} \in (\mathbb{R}^3)^{N \times M}$, where each entry $\mathcal{P}_{ij}$ is the set that describes the color in the same pixel position.

The way of obtaining the matrix $\mathcal{P}$ is natural, associating the value of the set which determines the corresponding pixel to each entry $\mathcal{P}_{ij}$. In order to increase the accuracy of the model, the most appropriate approach is to analyze the pixel neighborhood since the colour variation is gradual in an image without modifications. This feature known *a priori* and satisfied by images without modification is fundamental to our model, because based on that, we can determine if a pixel is correct using the concordance with his neighbors in the associated matrix.

This existing concordance between neighboring pixels is expressed as a correlation among them. This feature is known and widely used by several authors, as in (Julio et al., 2013.). The correlation calculation for each pixel of the image, according to image size, can be computationally expensive. Because of that, we propose to follow the same approach to evaluate statistical measures to determine concordance but trying to save the computational time.

To study the proximity of each pixel, which in this model is the immediate neighbors, we will introduce another matrix, this time of real components instead of vector components, denoted by $\mathcal{FP} \in \mathbb{R}^{N \times M}$, and where each element is defined by

$$\mathcal{FP}_{ij} = \mathcal{F}(\mathcal{P}_{ij}) \tag{1}$$

where $\mathcal{F}$ is a scalar function of the vector variable that measures the proximity of $\mathcal{P}_{ij}$ point with it nearest neighbors.

To illustrate the proximity of the pixel, in other words, their immediate neighbors, we present an example in Fig. 2.

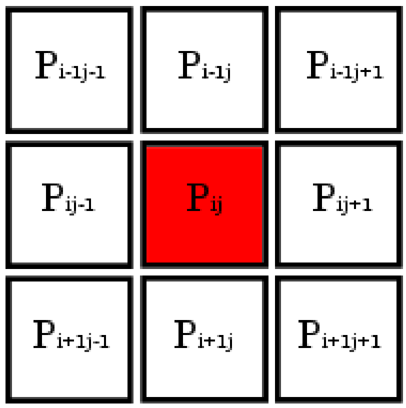


Figure 2: Immediate neighbor of a pixel.

We can observe it as a matrix, where each square represents a pixel. To enter the neighborhood idea, we will take the pixel in the middle, in red, to analyze. We see that the pixels around it make up what we call the immediate neighbors, this is, neighbors who are connected to it sideways, vertically and diagonally.

3.1 Arithmetic Mean

Arithmetic mean is a measure of proximity of each pixel, and can be defined as

$$\begin{aligned} \mathcal{F}(\mathcal{P}_{ij}) &= \frac{1}{8}\Big(\|\mathcal{P}_{ij} - \mathcal{P}_{i+1,j}\|^2 + \|\mathcal{P}_{ij} - \mathcal{P}_{i-1,j}\|^2 + \|\mathcal{P}_{ij} - \mathcal{P}_{i,j+1}\|^2 + \|\mathcal{P}_{ij} - \mathcal{P}_{i,j-1}\|^2 \\ &+ \|\mathcal{P}_{ij} - \mathcal{P}_{i-1,j-1}\|^2 + \|\mathcal{P}_{ij} - \mathcal{P}_{i-1,j+1}\|^2 + \|\mathcal{P}_{ij} - \mathcal{P}_{i+1,j-1}\|^2 + \|\mathcal{P}_{ij} - \mathcal{P}_{i+1,j+1}\|^2 \Big) \end{aligned} \tag{2}$$

3.2 Variance

A different choice for the proximity measurement is considered as the variance measure of proximity, replacing the definition of the function $\mathcal{F}$ given in Eq 3 as

$$
\begin{aligned}
\mathcal{F}(\mathcal{P}_{ij}) &= \frac{1}{8}\max_{k=1,2,3}\Big((\mathcal{P}_{i,j+1}[k]-\mu[k])^2 + (\mathcal{P}_{i,j-1}[k]-\mu[k])^2 \\
&+ (\mathcal{P}_{i+1,j}[k]-\mu[k])^2 + (\mathcal{P}_{i-1,j}[k]-\mu[k])^2 + (\mathcal{P}_{i-1,j-1}[k]-\mu[k])^2) \\
&+ (\mathcal{P}_{i-1,j+1}[k]-\mu[k])^2 + (\mathcal{P}_{i+1,j-1}[k]-\mu[k])^2 + (\mathcal{P}_{i+1,j+1}[k]-\mu[k])^2\Big)
\end{aligned} \tag{3}
$$

Notice that we are considering the variance of each component (R, G, B) that compose the pixel $\mathcal{P}_{ij}$, then, the proximity measure is the maximum of these variances.

3.3 Boundary Conditions

The above equations describe the processing of the interior points of the solution domain. It is necessary to establish conditions on the boundaries of this area in order to have a unique solution. The boundary conditions are imposed relations over the borders which usually relate the values of the variables on the border with the values inside. In the current context, the contour pixels, does not have eight neighbors, so we just renormalize the equations above.

3.4 Methodology

The proposed model can be summarized by the following items:

1. Get the associated matrix $\mathcal{P}$ through the target image;

The matrix $\mathcal{P}$ is formed of vector components, where each entry of matrix corresponds to one pixel of the image. This vector component is a vector of RGB intensities, i.e., we have a vector $v \in R^3$, wherein each component of this vector corresponds to the color intensity of R, G and B respectively. Each component, R, G and B can receive value from 0 to 255.

An example of a vector v is $(122, 67, 28)$, the pixel has 122 red intensity, 67 green intensity and 28 blue intensity, thereby forming its color.

2. Generate the proximity matrix $\mathcal{FP}$;

A new matrix should be generated to calculate the proximity of the pixel with its neighbors. The $\mathcal{FP}$ matrix is generated from the matrix $\mathcal{P}$ obtained in the previous step, $\mathcal{FP} = \mathcal{F}(\mathcal{P}_{ij})$, where F corresponds to a proximity function, we use the arithmetic mean and variance for this. We apply the proximity function in each entry (pixel) of matrix $\mathcal{P}$.

3. Classify each pixel of the image, with the associated matrix, as "correct" or "wrong";

Finally, we classify each pixel of the image analyzing $\mathcal{FP}$ matrix obtained in step 2. To classify, the value of each $\mathcal{FP}$ entry must be sufficiently small, i.e., must comply a criteria $\mathcal{F}(\mathcal{P}_{ij}) < \sigma_0$, where σ_0 corresponds to the average of the values reached by the matrix $\mathcal{FP}$.

4. If you get an incorrect sequence of pixels, it is likely that there is a message in that specific area.

4 IMPLEMENTATION

In order to infer knowledge from the studied data set, which will be presented later on, we implemented a software that is able to analyze and classify the pixels using the model described in Section 3.

Developed in the Python language, our software has three independent modules: learner, classifier and exhibition. The first one is responsible for the analysis phase, which consists in getting the associated matrix using the target image and generating the proximity matrix. The second module, on the other hand, performs the data classification itself, which is the prediction, by taking as input the enveloped proximity matrix from the first module and the instances whose outputs should be predicted. Finally, the third module returns the positions of the matrix where the incorrect pixels were found and also returns a visual result from it, an image with incorrect pixels highlighted.

The tool used in the image processing was *OpenCV - Open Source Computer Vision Library*, which was extremely important since it has a wide image processing module. Through it, it was possible to transform an input image into an output matrix. Another tool used was the Pylab, famous Python module that generates graphics and manipulates and analyzes images. In the current context, this module was used in the mathematical part of the proposed model.

5 NUMERICAL RESULTS

In this section, we first will explain how we get the data set used in this work and how the experiments were conducted. After this, the experimental results involving the different parameters and types of techniques will be presented and discussed. Initially, it is important to highlight that due to the large volume of results, we chose a graphical presentation technique and exposed some of the image results. The experiments were split into three groups: *(i)* computer drawings and design, *(ii)* B&W images and *(iii)* other images. For each group of experiments, fifty results were picked for a significant sample of results. Finally, all results are shown. Figure 3, 7 and 11 present the comparative graphics performance between the arithmetic mean and variance for each group of pictures.

5.1 Experiments

Data Set

One of the initial and most complex tasks in a work that depends on image records is the acquisition of the corresponding data set. In most cases, the construction of a specific and ideal one is impractical. Therefore, we used a data set we own, except for computer drawings and designs. This data set contains in total 900 records, divided into three different sets: grayscale images, computer drawings and design and images with various colors or elements. In each set there are images with different sizes and quality. This catalog was composed in this way to be a representative sample of images.

Computational Experiments

In the computational experiments performed, the tests were made using controlled experiments, i.e., using a cover image with certain part of the bits modified. The place of insertion of the message was defined using the *srand* function, feeding it with the index of the figure, and finally applying the mod operator using the number of rows and columns. This place is stored for later confer with the numerical output of the software.

The comparison is trivially, the numeric return is compared with the positions of the matrix that were stored, and this sets the model's accuracy rate for each image.

5.2 B&W Images

In the graphic shown in the fig. 3, we can see the effectiveness of the model using the two statistical measures described in section 3. The graph also shows comparative results obtained by both methods. It is important to note that the variance proved to be a more effective method than the arithmetic mean, but both showed satisfactory results for the objectives mentioned above. We also noted that in some rare cases, the arithmetic mean was more effective than the variance.

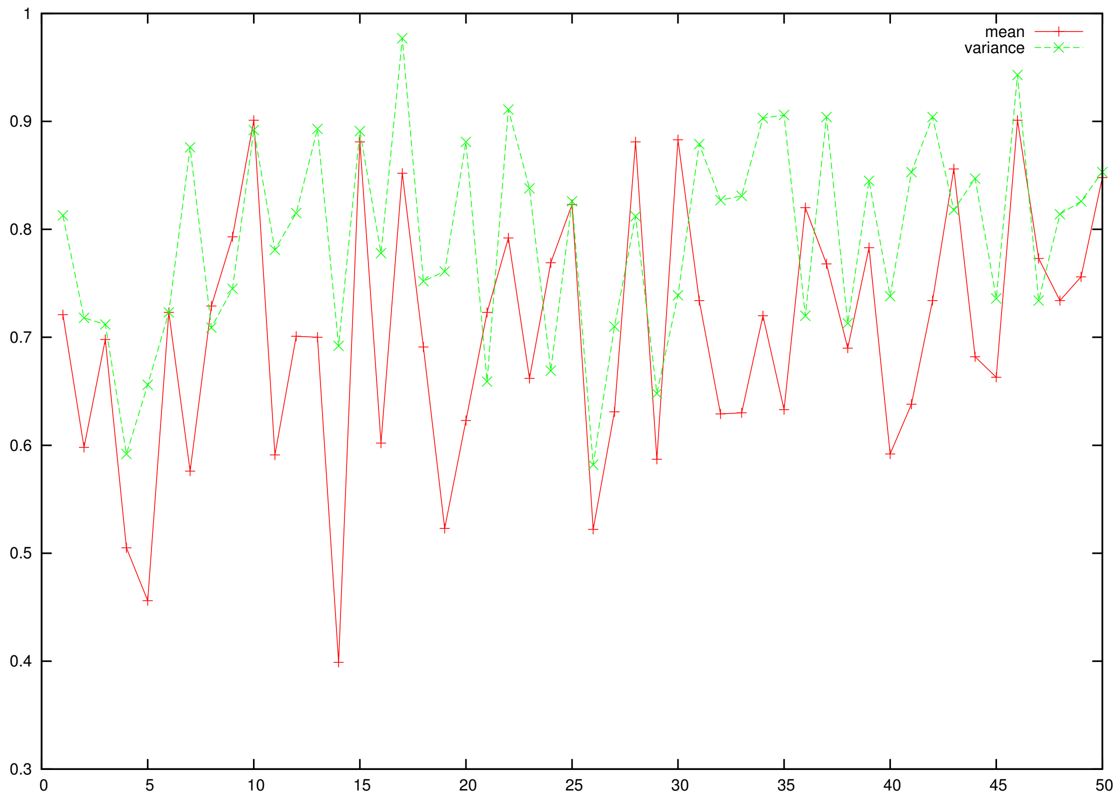
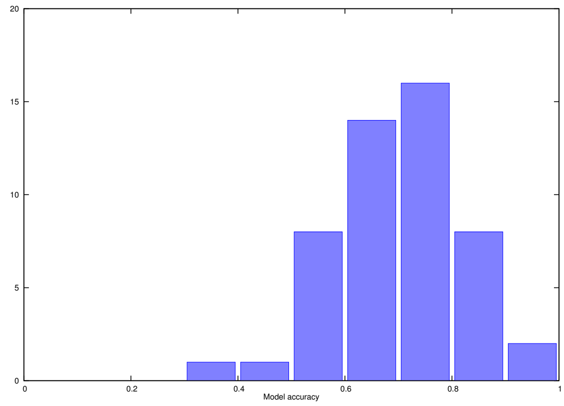
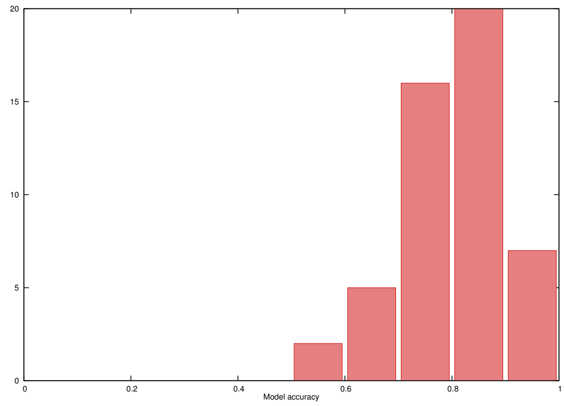


Figure 3: Hit rate of both measures.

Histograms shown in fig. 4 show the hit rate of each statistical measure. The hit rate of arithmetic mean was 70.2% and of variance was 79.2%. We can observe that variance's histogram is shifted to the right in relation to the arithmetic mean, suggesting that variance is the best method.



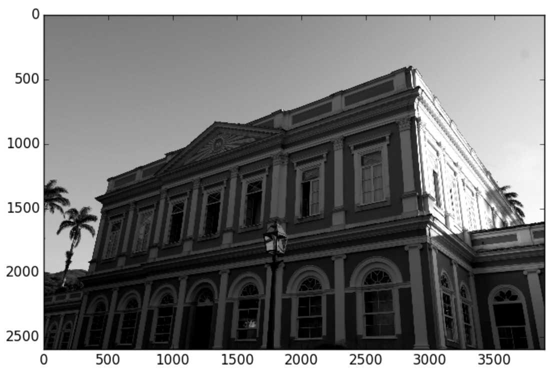
(a) Histogram of arithmetic mean



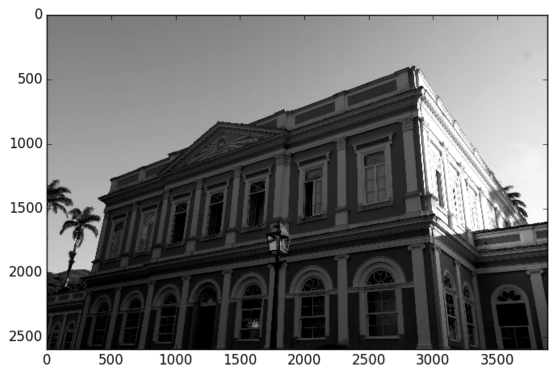
(b) Histogram of variance

Figure 4: Histogram of arithmetic mean and variance.

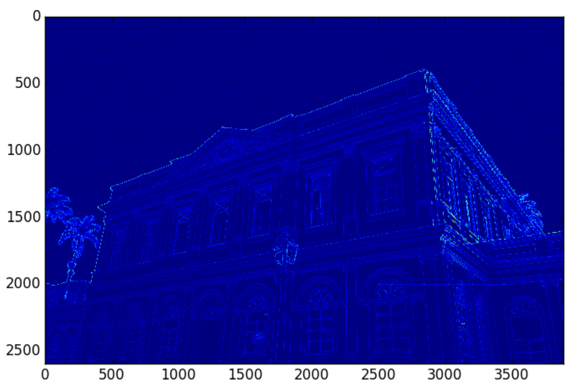
In the fig. 5 and 6 we will show some of the visual results. It is important to note that the numeric return of the model is the most significant for the analysis of results. Figure 5 shows the image (a) before and (b) after receiving steganography, we note that there is no visual difference between them.



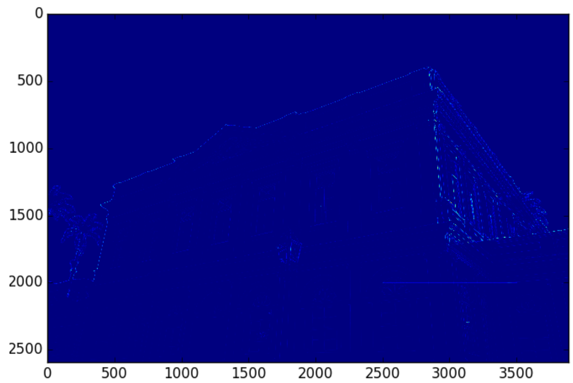
(a) Cover-image



(b) Stego-image



(c) Visual result using arithmetic mean

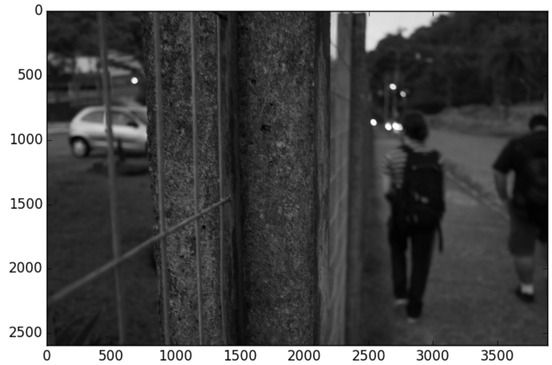


(d) Visual result using variance

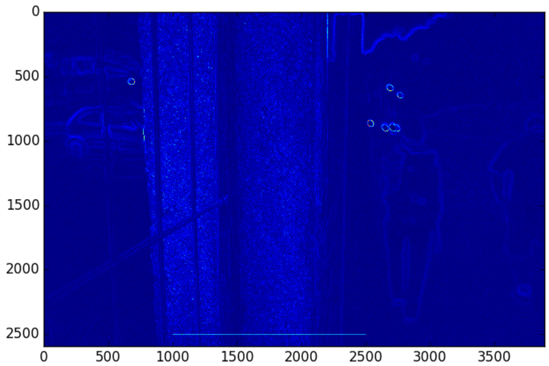
Figure 5: Figures (c) and (d) show the results obtained after apply the model on stego-image (b).

It is possible to note that the variance elimate the noise of the image and shows more efficiently sequences of wrong pixels. Also, it is important to note that the borders (or edges) of the image are classified as incorrect. There is a subtle but visible sequence of pixels between columns 2500 e 3500 in the bottom of the image. On the other hand, in fig 6, the sequence of pixels can be seen more clearly on the bottom of image.

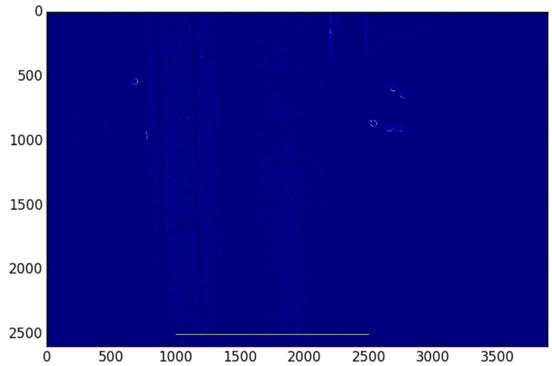
(a) Cover-image



(b) Stego-image



(c) Visual result using arithmetic mean



(d) Visual result using variance

Figure 6: Figures (c) and (d) show the results obtained after apply the model on stego-image (b).

5.3 Computer Drawings and Design

In the graphic shown in the fig. 7, we can see the effectiveness of the model using the two statistical measures described in section 3. The graph also shows comparative results obtained by both methods.

Similarly to the B&W, variance achieved better results than the arithmetic mean. However, arithmetic mean is also a excelent statistical measure to. It is important to observe that there are some cases where arithmetic mean achieved better results than the variance, and also some similar results.

It is also important to observe that the proposed mathematical model was easier to deal with computer drawings that with B&W. This fact is supported by (Johnson and Jajodia, 1998), where it is said that grayscale images are better candidates to receive steganography.

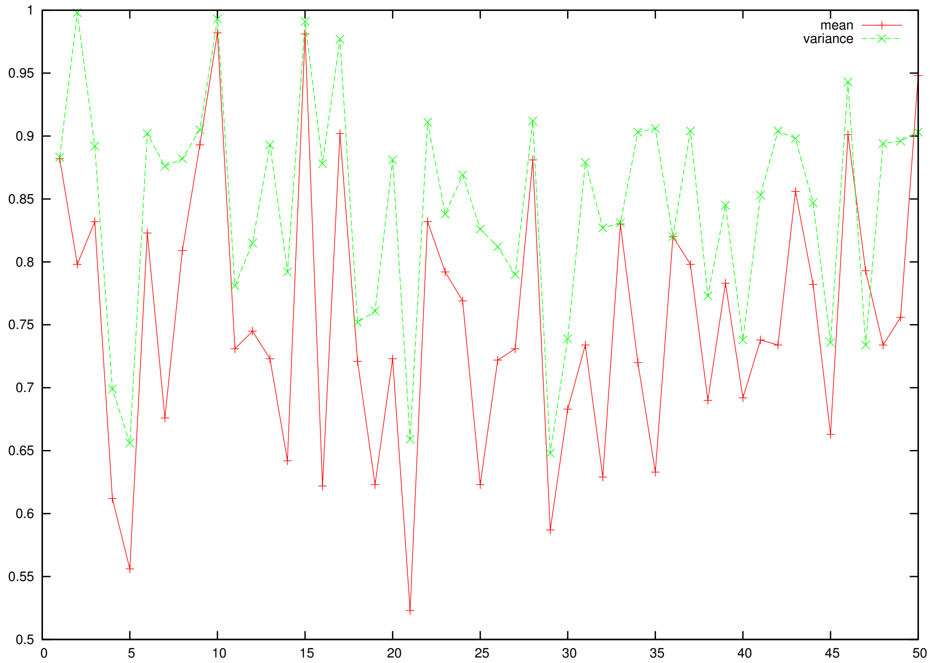
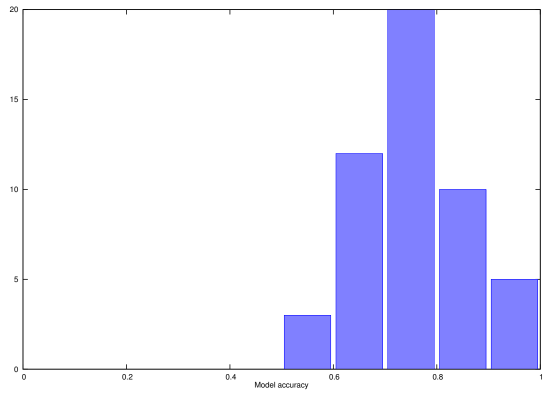
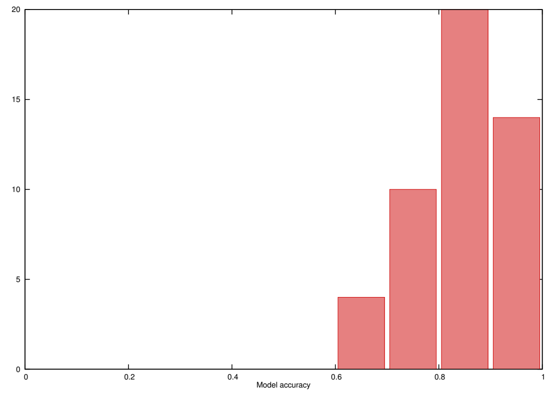


Figure 7: Hit rate of both measures.

Histograms shown in fig. 8 show the hit rate of each statistical measure. The hit rate of arithmetic mean was 75.5% and of variance was 84.4%. We can observe that the histogram behavior is similar to the previous case, variance's histogram is shifted to the right in relation to the arithmetic mean.



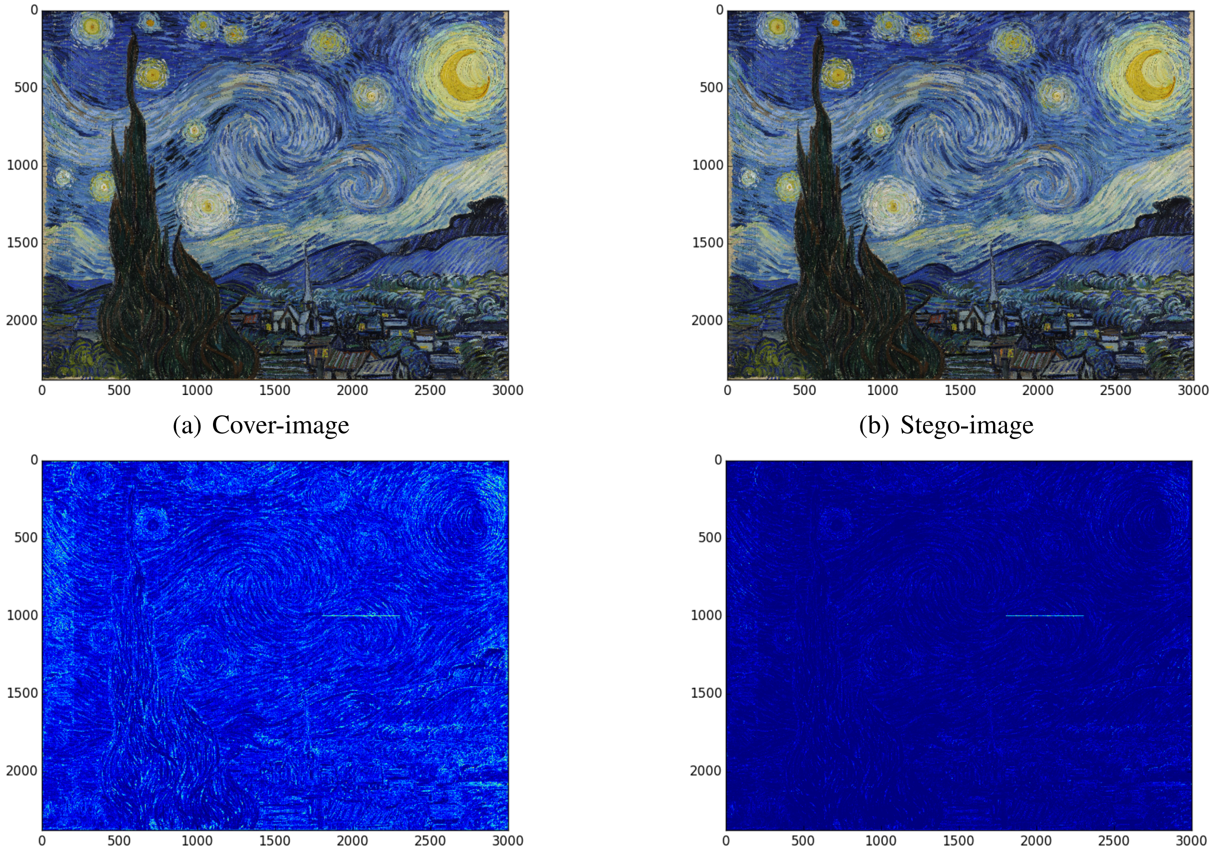
(a) Histogram of arithmetic mean



(b) Histogram of variance

Figure 8: Histogram of arithmetic mean and variance.

Figure 9 show the work *De sterrennacht* (1889) by Vincent van Gogh.



(a) Cover-image (b) Stego-image

(c) Visual result using arithmetic mean (d) Visual result using variance

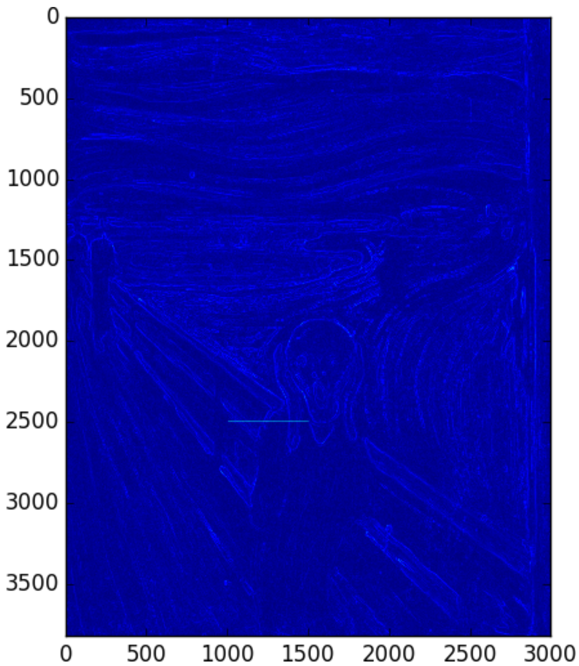
Figure 9: Figures (c) and (d) show the results obtained after apply the model on stego-image (b).

It is possible to see a disconnected line on the right side amidst the image noise.

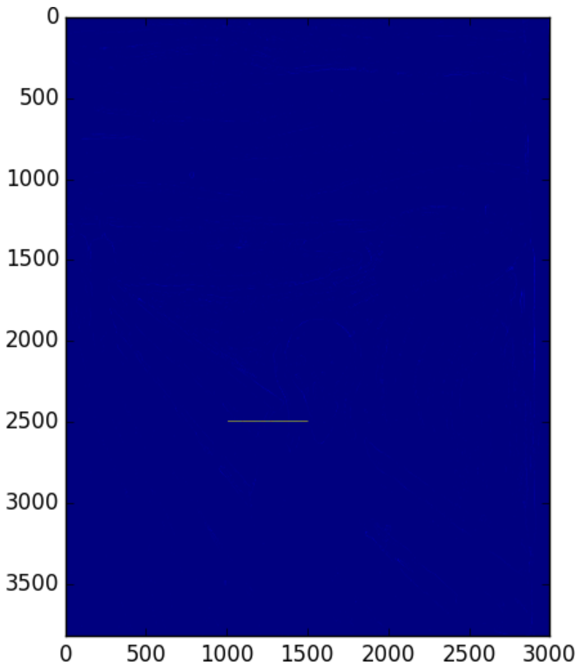
Figure 10 show the work *Skrik* (1893) by Edvard Munch.

(a) Cover-image

(b) Stego-image



(c) Visual result using arithmetic mean



(d) Visual result using variance

Figure 10: Figures (c) and (d) show the results obtained after apply the model on stego-image (b).

In this case, because of the image dynamic, the mathematical model was capable to eliminate most of the noise and this made a more clear result.

5.4 Other images

In the graphic shown in the fig. 11, we can see the effectiveness of the model using the two statistical measures described in section 3. The graph also shows comparative results obtained by both methods. It is important to note that, like the previous groups, the variance proved to be a more effective method. Also, this group showed an improvement over B&W figures and a decline in comparsion to drawings. It has also cases where arithmetic mean obtained similar or better results than variance.

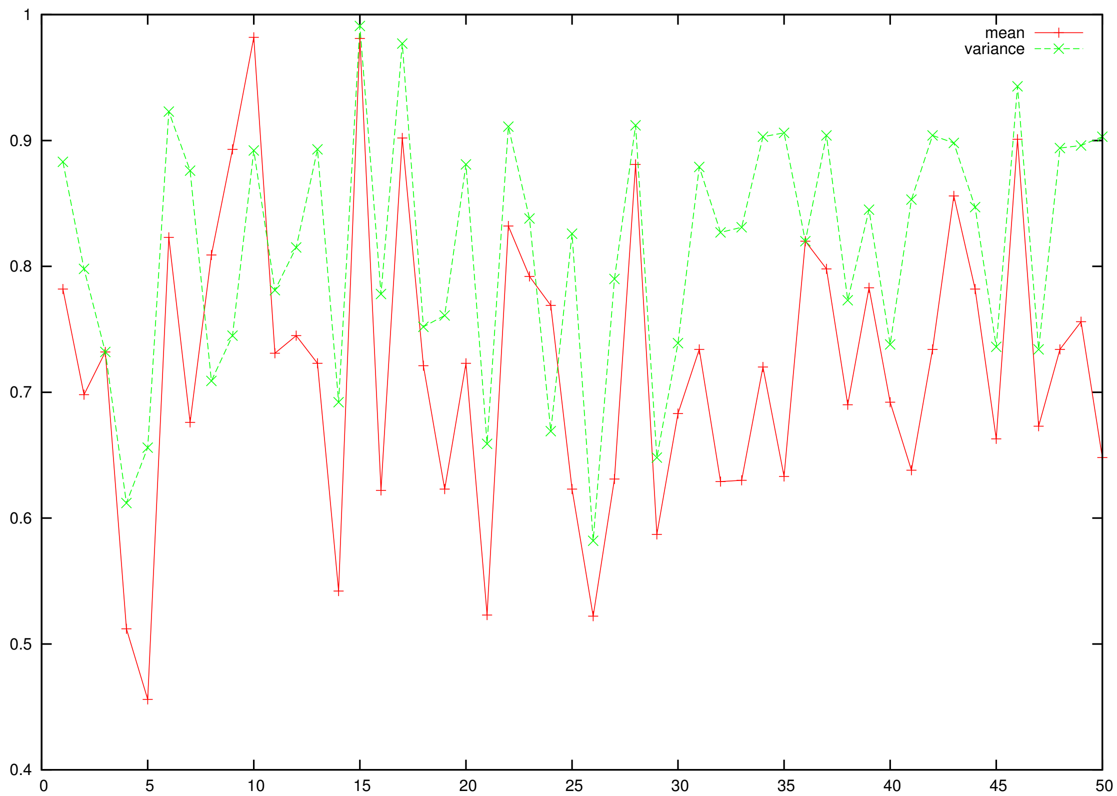
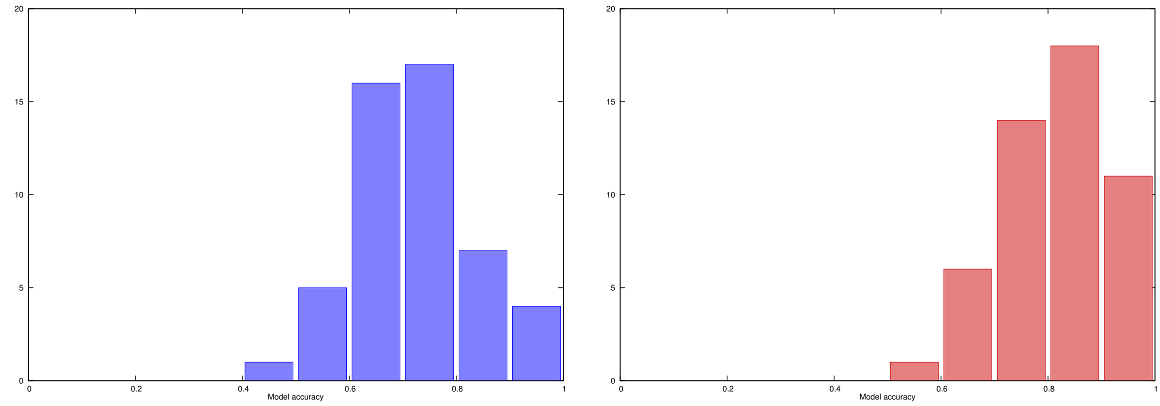


Figure 11: Hit rate of both measures.

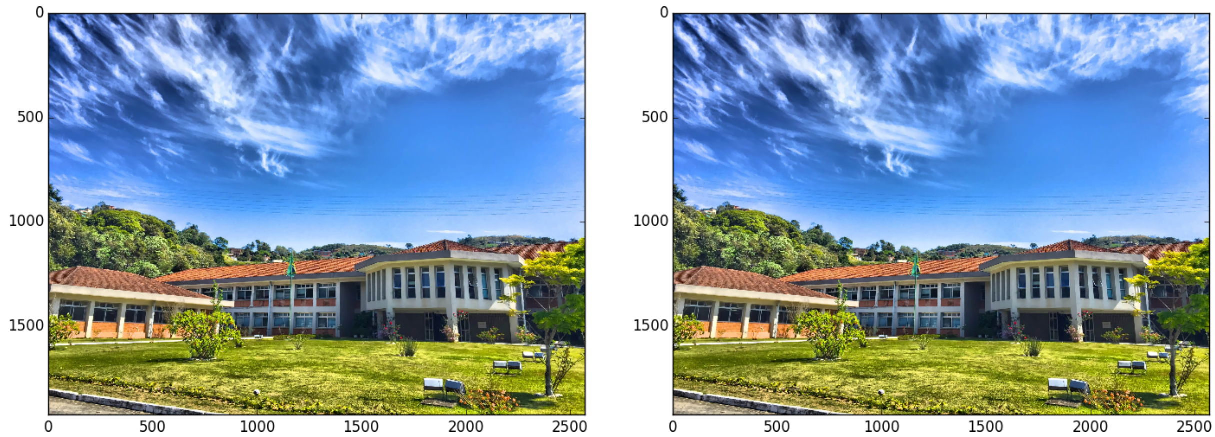
Histograms shown in fig. 12 show the hit rate of each statistical measure. We can see that both methods are effective. The hit rate of arithmetic mean was 72% and of variance was 81.5%. Also, we can observe that variance's histogram is lightly shifted to the right in relation to the arithmetic mean, corroborating the superiority of this method.



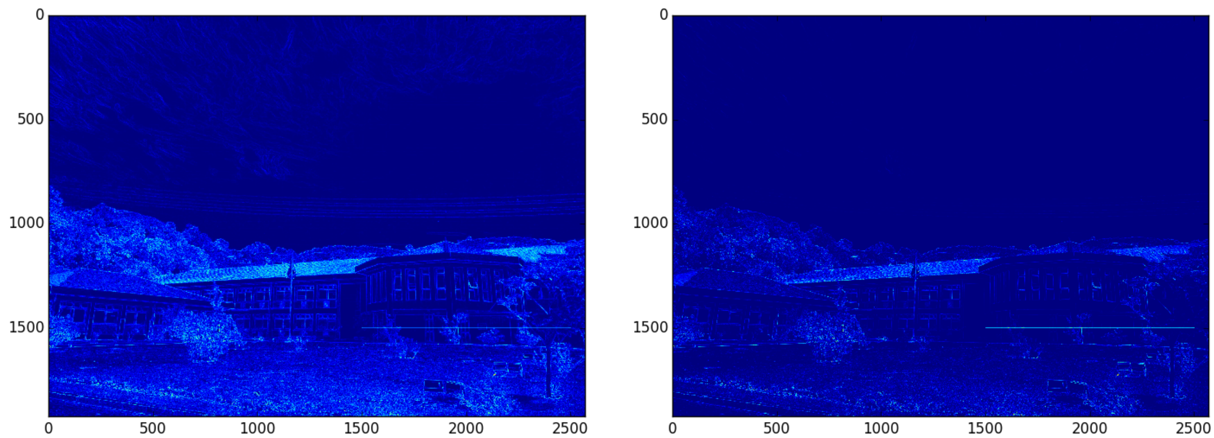
(a) Histogram of arithmetic mean (b) Histogram of variance

Figure 12: Histogram of arithmetic mean and variance.

Figures 13 and 14 show some of the visual results. First case shows a thin line between columns 1500 and 2000, at line 1500. In the other case, the result is evident in the middle of image.

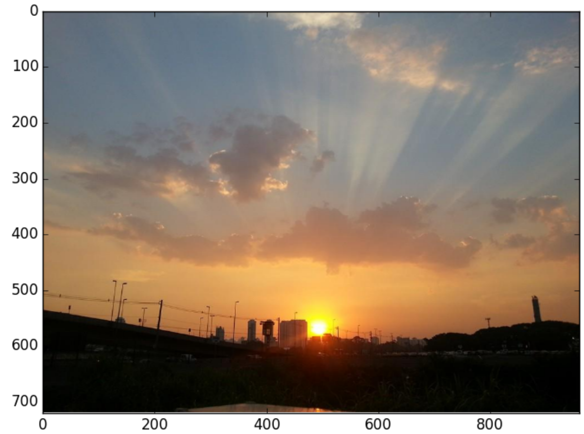


(a) Cover-image (b) Stego-image

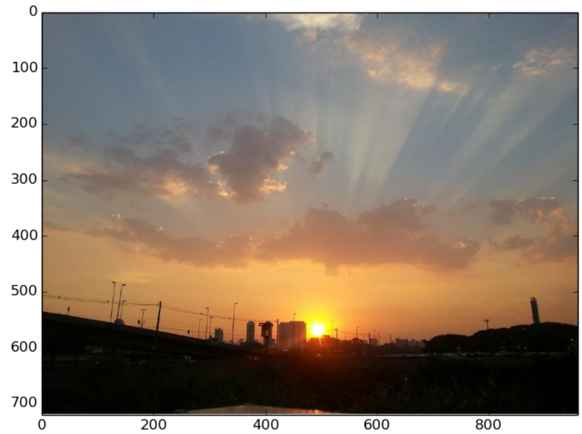


(c) Visual result using arithmetic mean (d) Visual result using variance

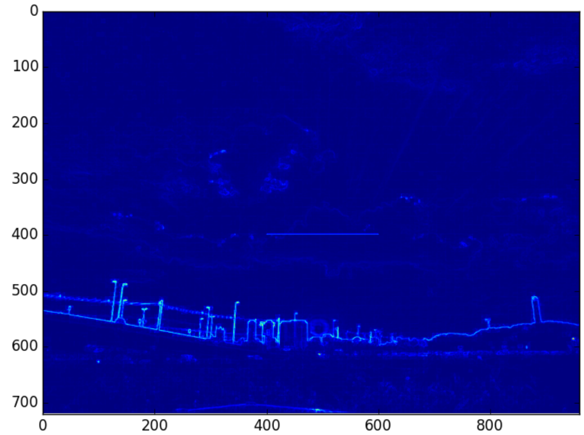
Figure 13: Figures (c) and (d) show the results obtained after apply the model on stego-image (b).



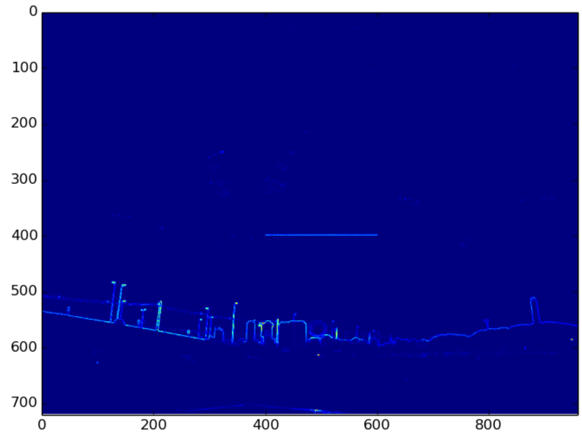
(a) Cover-image



(b) Stego-image



(c) Visual result using arithmetic mean



(d) Visual result using variance

Figure 14: Figures (c) and (d) show the results obtained after apply the model on stego-image (b).

6 CONCLUSIONS AND FUTURE WORK

This paper presented a new mathematical model that predicts steganography on images. After exposing the basic concepts about the fundamentals shown in this work, we conclude that using pixel classification is an effective ally in the study of prediction of steganography in images. The classification pixel technique was incorporated to a mathematical model, and for this, we used two proximity measures: arithmetic mean and variance.

Therefore, in addition to theoretically demonstrate the technique and justify their choice, a series of experiments was held on three differente groups of image: B&W figures, computer drawings and images with various colors or elements. After a round of experiments, it is concluded that variance proved to be a measure of proximity more effective although arithmetic mean have also shown satisfactory results.

We can also conclude that use proximity measures provides a characterization appropriate to the objectives, being able to distinguish between areas of an image where there was a change in pixel values.

Among the next goals, we can highlight the incorporation of a filter capable of detect the borders of the image, in order to better characterize the pixels and not let thus to influence at the time of classification. Finally, to use a proximity measure that increases the accuracy of the model and provides a great solution.

For all these reasons, this work helps to explain the technique used and demonstrate its effectiveness with respect to the proposed theme. It is important to note that we were unable to find other studies that aim prediction of steganography using a mathematical model based on proximity of pixel classification. Therefore, the present work opens up possibilities for further research in this area.

7 ACKNOWLEDGMENTS

The authors would like to thank the support from CNPq.

REFERENCES

Tony F Chan and Jianhong Jackie Shen, 2005. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. Siam.

Guillermo Francia, Ming Yang, Monica Trifas, et al. Applied image processing to multimedia information security. In *Image Analysis and Signal Processing, 2009. IASP 2009. International Conference on*, pages 104–107. IEEE, 2009.

Jessica Fridrich, Miroslav Goljan, and Rui Du. Reliable detection of lsb steganography in color and grayscale images. In *Proceedings of the 2001 workshop on Multimedia and security: new challenges*, pages 27–30. ACM, 2001.

Jiri Fridrich and M Long. Steganalysis of lsb encoding in color images. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 3, pages 1279–1282. IEEE, 2000.

Neil F Johnson and Sushil Jajodia, 1998. Exploring steganography: Seeing the unseen. *Computer*, vol. 31, n. 2, pp. 26–34.

E.P. Julio, W. G. Brazil, and C.V.N. Albuquerque. *VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais.*, chapter Esteganografia e suas Aplicações. 2013.

Tayana Morkel, Jan HP Eloff, and Martin S Olivier. An overview of image steganography. In *ISSA*, pages 1–11, 2005.

Azriel Rosenfeld, 2013. *Multiresolution image processing and analysis*, volume 12. Springer Science & Business Media.

Joshua Silman, 2001. Steganography and steganalysis: an overview. *SANS Institute*, vol. 3, pp. 61–76.

Andreas Westfeld and Andreas Pfitzmann. Attacks on steganographic systems. In *Information Hiding*, pages 61–76. Springer, 2000.

Michael Whitman and Herbert Mattord, 2011. *Principles of information security*. Cengage Learning.