Análise Comparativa de Métodos para Esteganografia Digital em Imagens

Diego H. B. Zanchett
diego.zanchett@aluno.cefet-rj.br
CEFET/RJ–*Campus* Petrópolis

Jurair R. P. Junior
jurair.junior@cefet-rj.br
CEFET/RJ–*Campus* Petrópolis

Diego B. Haddad
diego.haddad@cefet-rj.br
CEFET/RJ–*Campus* Petrópolis

Laura S. Assis
laura.assis@cefet-rj.br
CEFET/RJ–*Campus* Petrópolis

ABSTRACT

Sensitive information being shared on the internet is growing. Because of this, it is increasingly necessary to take security measures whilst this information travels in the network. Digital steganography allows one to send sensitive information in a hidden manner. Although there is a plethora of techniques for such a goal, finding an appropriate one is not always simple. This paper implements and compares spatial-domain digital steganography techniques in both RGB and grayscale images. A frequency-domain heuristic for reducing the visual impact of digital steganography in grayscale images is presented. As another result of this work, a dataset is also available in the Kaggle platform with 18 GB of images, containing secret messages using the techniques under study. In addition, a Python language library was also made available in the PyPI repository, allowing for both concealment and revelation of messages using the presented digital steganography methods.

KEYWORDS

Estenografia Digital, Segurança da Informação, Processamento Digital de Imagem

1 INTRODUÇÃO

O advento da internet engendrou diversos avanços tecnológicos em ambientes acadêmicos e corporativos, com repercussões possivelmente ainda mais drásticas em outras áreas da sociedade moderna. Paralelamente a tais avanços, há uma preocupação crescente com possíveis violações de privacidade, as quais motivam o desenvolvimento de métodos eficazes de proteção à informação digital. Como métodos promissores pertencentes a tal espectro tecnológico, cumpre realçar as técnicas de esteganografia digital, sobre as quais concentra-se este artigo.

A esteganografia é uma técnica que consiste em ocultar uma mensagem no interior de outra mensagem, de sorte que apenas o remetente e o destinatário da mensagem, os quais possuem conhecimento da existência da informação escondida, tenham acesso a tal informação. A palavra "esteganografia" deriva do grego *steganós* (que significa oculto, escondido) e *grafia*, que significa escrita [1]. Na sua vertente digital, a esteganografia se concentra em informações armazenadas em meios digitais [2]. O problema da esteganografia pode ser interpretado como uma variante do problema do prisioneiro, no qual dois indivíduos almejam encetar uma comunicação por um meio sem levantar suspeitas de eventuais intrusos [3].

Descabe confundir a esteganografia com a criptografia, pois enquanto esta se destina à codificação de uma mensagem, aquela almeja camuflar uma mensagem no interior de informações outras, geralmente de caráter áudio-visual. Na eventualidade de tal inserção de informações ser descoberta por um interceptador, este poderá facilmente resgatar a imagem original, caso a técnica utilizada pela esteganografia seja por ele conhecida. Ressalte-se que, neste contexto, o interceptador não precisa conhecer nenhuma chave de descriptografia específica [4]. A incorporação da mensagem nos diferentes métodos de esteganografia deve reduzir a distorção resultante, podendo ser formulada como um problema de codificação de fonte com um critério de fidelidade [3]. Caso seja conveniente garantir maior segurança à informação transmitida ou armazenada, técnicas de criptografia podem ser simultaneamente combinadas com as de esteganografia.

A utilização de meios digitais para realizar a comunicação e o armazenamento de informações tem aumentado gradativamente com o tempo, assim como a quantidade de ataques que visam roubar tais informações [5]. Com este cenário, torna-se cada vez mais necessário garantir a proteção e integridade dos dados, sendo crescente o desenvolvimento de tecnologias de criptografia e esteganografia com o objetivo de evitar que pessoas não autorizadas tenham acesso aos mesmos [4, 6].

É importante ressaltar que técnicas de esteganografia digital podem ser empregadas em ataques maliciosos para evitar que o vazamento de dados sensíveis seja percebido por ferramentas *anti-malware*. Nestes ataques, os dados sensíveis são escondidos pelo atacante nas informações que já trafegariam normalmente pela rede. Deste modo, tais dados se tornam invisíveis para os sistemas de segurança.

Dentre as principais contribuições deste artigo, cumpre elencar:

(1) Propõe uma modificação na técnica baseada em SSB-4;
(2) Comparação entre diversas técnicas de esteganografia digital presentes na literatura, a saber:
   - LSB (*Least Significant Bit*);
   - LSB Escala de Cinza;
   - SSB-4 (*System of Steganography Using Bit 4*);
   - SSB-N (*System of Steganography Using Bit N*);
   - Esteganografia baseada em DCT (*Discrete Cossine Transform*);
   - Esteganografia baseada em FFT (*Fast Fourier Transform*).
(3) Comparação da técnica LSB tanto no domínio RGB quanto em escala de cinza.

As técnicas de esteganografia são comumente classificadas em domínio espacial ou domínio da frequência. As técnicas no domínio espacial alteram diretamente os bits menos significativos dos valores dos pixels que compõem a representação matricial de uma imagem [7]. Dentre as técnicas apresentadas neste trabalho, as seguintes estão contidas no domínio espacial: LSB, LSB escala de cinza, SSB-4 e SSB-N. Já as técnicas contidas no domínio da frequência convertem a imagem de capa para o domínio da frequência aplicando uma transformada antes de armazenar a mensagem secreta [7]. Dentre as transformadas que podem ser aplicadas, podemos citar a transformada do cosseno discreto, transformada discreta de Fourier, transformada *wavelet*, entre outras [8]. As técnicas citadas neste trabalho que se encontram neste domínio são a esteganografia baseada em DCT e a esteganografia baseada em FFT. Uma apresentação mais detalhada sobre cada uma destas técnicas é realizada na Seção 2.

Experimentos computacionais foram realizados para comparar as técnicas implementadas, assim como avaliar as modificações propostas. Os resultados obtidos mostram que após o processo de esteganografia a imagem que contém a mensagem secreta apresenta pouca distorção visual. Além disso, para as técnicas DCT e FFT é apresentada uma baixa taxa de erro na mensagem secreta após sua recuperação. É importante ressaltar que para as demais técnicas citadas neste trabalho, em condições ideais e desde que a imagem de capa não passe por processos de compactação ou edição, não são observados erros na recuperação das mensagens secretas.

Este artigo está organizado da seguinte maneira. Na Seção 2 é apresentado o referencial teórico das técnicas de esteganografia digital em foco, assim como a metodologia proposta, bem como as modificações realizadas nas técnicas DCT e FFT. Os principais trabalhos relacionados a este presentes na literatura são abordados na Seção 3. Os resultados provenientes dos experimentos computacionais e uma comparação das técnicas em estudo são apresentados e discutidos na Seção 4. A Seção 5 contém as considerações finais e trabalhos futuros.

2 ABORDAGEM TEÓRICA E METODOLOGIA

Uma comparação entre seis técnicas de esteganografia digital será efetuada neste artigo. Estas técnicas são baseadas nos algoritmos de: *(i)* Bit Menos Significativo (*Least Significative Bit* - LSB), *(ii)* Bit Menos Significativo em Imagens em Escala de Cinza (*Least Significative Bit Grayscale* - LSB Grayscale), *(iii)* Sistema de Esteganografia Utilizando Bit 4 (*System of Steganography Using Bit 4* - SSB-4), *(iv)* Sistema de Esteganografia Utilizando Bit Pseudo-Aleatório (*System of Streganography Using Random Bit* - ssb-n), *(v)* Transformada do Cosseno Discreto (*Discrete Cossine Transform* - DCT), e *(vi)* Transformada Rápida de Fourier (*Fast Fourier Transform* - FFT).

Tais técnicas foram implementadas utilizando a linguagem *Python*. Foram utilizados dois *datasets* extraídos da plataforma Kaggle, um *dataset* de imagens (Landscape Pictures) [1], que foi utilizado como imagem pública, também conhecido como imagem de capa, para armazenar as mensagens secretas, e um *dataset* de artigos da Wikipedia (*English Wikipedia Articles* 2017-08-20 *SQLite*) [2] que foi utilizado para compor as mensagens secretas. As técnicas comparadas serão descritas a seguir.

2.1 Bit Menos Significativo

As abordagens mais comuns para realização de inserção de mensagens em imagens usando ruído são baseadas na técnica LSB (*Least Significant Bit*), que é um método de esteganografia não adaptativo no domínio espacial, o qual mobiliza os bits menos significativos para armazenar os dados que se pretende proteger. A Figura 1 ilustra a utilização da técnica LSB através da representação de cada pixel da imagem. Nela pode-se observar que os bits sinalizados em cinza estão representando a mensagem que está sendo armazenada, enquanto que os demais são referentes às informações da imagem original.

Assim, é possível selecionar um bit menos significativo em cada byte da imagem como local onde se esconderá a mensagem, de forma segura, sem causar alterações que sejam perceptíveis visualmente na mesma [9, 10]. Neste artigo, diversos testes foram realizados, e observou-se empiricamente que alterações nos dois últimos bits da imagem proporcionam a melhor relação entre minimização da distorção visual e maximização do espaço para armazenamento da mensagem [11]. Portanto, este artigo utiliza os dois bits menos significativos de um conjunto de oito bits que representam os valores de cada canal (vermelho, verde e azul) de cada pixel da imagem, como representado na Figura 1.

	Canal Vermelho (R)								Canal Verde (G)								Canal Azul (B)							
1º Pixel	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1	0	1	0
2º Pixel	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1
3º Pixel	1	0	1	1	0	0	0	1	0	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
nº Pixel	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	0	0	0	1	1	1	0	1	0

Figura 1: Representação de cada pixel da imagem utilizando a técnica LSB.

A mensagem é codificada neste trabalho por meio da codificação ASCII. Os bits menos significativos da imagem são substituídos pelos bits da mensagem que se deseja proteger. O Algoritmo 1 apresenta o pseudocódigo da técnica LSB.

O algoritmo inicia recebendo a imagem de capa (i_c) e a mensagem secreta (*msg*). Em seguida são extraídos os bits da imagem de capa através da função $getBits(i_c)$ (linha 1). A quantidade de bits da imagem de capa é obtida através da função $getSizes(i_c)$ (linha 2). A partir da quantidade de bits da imagem de capa calcula-se o tamanho máximo que a mensagem secreta pode ter, para que seja possível escondê-la dentro da imagem de capa. Este tamanho máximo é obtido como mostrado na linha 3, já que utilizaremos apenas um quarto dos bits para armazenar a mensagem secreta. Se for possível armazenar a mensagem secreta dentro da imagem de capa (linha 4), é executado um *loop* que realiza este procedimento (linhas 7 - 11). Este *loop* itera sobre o vetor que contém os bits da mensagem secreta ($bitsMsg$), extraindo dois bits por iteração. A cada iteração, estes dois bits são armazenados nos dois últimos bits de um byte da imagem de capa (linhas 8 - 9). Após armazenar os bits adiciona-se 8 à variável $contadorBits$, de modo a permitir o processamento do próximo byte (linha 10). Esta variável é um índice para o vetor i_{bits}. Após encerrar o *loop*, a variável i_m recebe a imagem contendo a mensagem secreta (linha 12). O algoritmo

[1] www.kaggle.com/arnaud58/landscape-pictures
[2] www.kaggle.com/jkkphys/english-wikipedia-articles-20170820-sqlite

termina retornando a imagem original devidamente modificada pela inserção da mensagem escondida (linha 13). O algoritmo 1 é executado em um tempo $O(n)$, onde n representa a quantidade de bits da mensagem secreta.

Algoritmo 1: : LSB (i_c, msg)

```
1:  i_bits ← getBits(i_c)
2:  size_ibits ← getSize(i_bits)
3:  tamMaxMsg ← ⌊size_ibits/4⌋
4:  if getSize(msg) ≤ tamMaxMsg then
5:     bitsMsg[ ] ← getAsciiBits(msg)
6:     contadorBits ← 7
7:     for each {bit_1, bit_2} ∈ bitsMsg do
8:        i_bits[contadorBits] ← bit_1
9:        i_bits[contadorBits + 1] ← bit_2
10:       contadorBits += 8
11:    end for
12:    i_m ← bitsToImage(i_bits)
13:    return i_m
14: end if
```

2.2 Bit Menos Significativo em Escala de Cinza

	Canal Escala de Cinza							
1º Pixel	1	0	1	1	0	1	1	1
2º Pixel	0	1	1	0	1	0	1	0
3º Pixel	1	0	1	1	0	0	0	1
...	...	...	...	...	...	...	...	...
nº Pixel	0	1	1	1	0	0	1	0

Figura 2: Representação de cada pixel da imagem utilizando a técnica LSB em Escala de Cinza.

A técnica LSB em escala de cinza realiza todo o processo mencionado na Seção 2.1, porém utilizando imagens em escala de cinza. Neste trabalho convertemos as imagens do *dataset* de testes para escala de cinza. A utilização de imagens em escala de cinza tem como objetivo reduzir tanto as distorções visuais na imagem pós-processada, quanto melhorar os valores obtidos nas métricas de avaliação utilizadas, *(i) Mean Squared Error* (MSE) e *(ii) Peak Signal to Noise Ratio* (PSNR).

Esta técnica é apresentada na Figura 2, na qual cada linha representa um pixel da matriz da imagem, percorrendo a respectiva matriz da imagem da esquerda para direta, linha a linha. Nesta figura, os bits na cor branca representam os bits de cada pixel que ficam inalterados no final do procedimento e na cor cinza estão representados os bits que são alterados para armazenar a mensagem secreta. Neste artigo, a comparação é realizada entre a imagem em escala de cinza original e a imagem em escala de cinza que contém a mensagem secreta. A conversão de RGB para escala de cinza foi realizada obedecendo a recomendação *ITU-R Recommendation BT.709*.

	Canal Escala de Cinza							
1º Pixel	1	0	1	1	0	1	1	1
2º Pixel	0	1	1	0	1	0	1	0
3º Pixel	1	0	1	1	0	0	0	1
...	...	...	...	...	...	...	...	...
nº Pixel	0	1	1	1	0	0	1	0

Figura 3: Representação de cada pixel da imagem utilizando a técnica SSB-4.

2.3 4º Bit (SSB-4) em Escala de Cinza

Em algumas das técnicas de esteganografia digital presentes na literatura são utilizados os bits menos significativos de uma imagem para armazenar a mensagem secreta. Porém, visando dificultar detecção da esteganografia digital através da análise dos dois últimos bits de cada pixel da imagem, a técnica (SSB-4) é baseada em armazenar a mensagem secreta no quarto bit da imagem de capa [12].

Nesta técnica, a imagem de capa é dividida em n partes iguais, destinando-se um *pixel* de cada parte para o armazenamento da mensagem secreta. O valor de n é calculado através da Equação (1):

$$n = \lfloor getSize(msg) \rfloor, \tag{1}$$

onde i_c representa a imagem de capa e msg representa a mensagem secreta.

Nesta técnica, após inserir a mensagem secreta no quarto bit dos pixels selecionados, é realizada uma normalização dos bits 1, 2, 3, e/ou 5 para que a diferença entre o valor original daquele pixel e o valor após a modificação seja mínima. Com isto espera-se que a distorção visual gerada por esta técnica seja mínima.

Esta técnica é apresentada na Figura 3, na qual cada linha representa um pixel da matriz da imagem sendo percorrida da esquerda para a direita, linha a linha. Enquanto os bits representados na cor branca são preservados durante o procedimento, os bits assinalados em cinza são alterados para incorporar a mensagem secreta. Por sua vez, os bits representados na cor amarela são modificados pela rotina de normalização, visando reduzir a diferença entre o pixel que contém a mensagem secreta e o pixel original, bem como reduzir a distorção gerada na imagem que conterá a mensagem secreta.

2.4 Bit Aleatório (SSB-N) em Escala de Cinza

	Canal Escala de Cinza							
1º Pixel	1	0	1	1	0	1	1	1
2º Pixel	0	1	1	0	1	0	1	0
3º Pixel	1	0	1	1	0	0	0	1
...	...	...	...	...	...	...	...	...
nº Pixel	0	1	1	1	0	0	1	0

Figura 4: Representação de cada pixel da imagem utilizando a técnica SSB-N.

Visando dificultar tanto a detecção quanto a extração da mensagem secreta armazenada em uma imagem na qual foi aplicada a

esteganografia digital, propomos uma nova abordagem baseada na técnica SSB-4. A abordagem proposta armazena cada bit referente à mensagem secreta em um bit escolhido aleatoriamente entre o primeiro e o quarto bit de cada pixel da imagem de capa. Para redução da discrepância entre o pixel da imagem modificada e o respectivo pixel da imagem de capa, são permitidas alterações dos demais bits situados entre o primeiro e o quinto bits.

Como a escolha do bit a ser alterado será aleatória, almeja-se gerar uma distorção na imagem resultante menos severa do que a engendrada pela técnica SSB-4, já que em alguns casos o bit a ser modificado será menos significativo do que o quarto bit.

É importante ressaltar que como a escolha do bit que conterá a mensagem secreta em cada pixel será aleatória, será necessário conhecer a sequência de índices utilizados para recuperar a mensagem secreta. Isto gera uma proteção adicional para a mensagem que está sendo armazenada na imagem de capa. Como definimos anteriormente um tamanho máximo para a imagem de capa, esta sequência de índices pode ser reutilizada em várias imagens, simplificando o processo de recuperação da mensagem secreta.

Visando reduzir ainda mais a distorção gerada ao armazenar a mensagem secreta na imagem de capa utilizado a técnica SSB-N, a imagem de capa é dividida em n partes iguais, de modo que é utilizado o primeiro pixel de cada parte para armazenar a mensagem secreta. O valor de n é calculado através da Equação (2), e o tamanho de n em bits é calculado através da Equação (3):

$$n = \lfloor getSize(msg) \rfloor, \tag{2}$$

$$getSize(n) = \left\lfloor \frac{getSize(i_c)/8}{getSize(msg)} \right\rfloor, \tag{3}$$

onde i_c representa a imagem de capa, msg representa a mensagem secreta e a função $getSize$ retorna o tamanho em bits. O tamanho da imagem de capa é dividido por 8, pois somente um bit de cada pixel é utilizado para armazenamento da mensagem escondida. Com isto se torna necessário conhecer o tamanho da mensagem secreta para sua recuperação, proporcionando mais segurança a esta técnica.

Esta técnica é apresentada na Figura 4, na qual cada linha representa um pixel da matriz da imagem sendo percorrida da esquerda para a direita, linha a linha. As cores de cada bit seguem as mesmas regras utilizadas na apresentação da Figura 3. Estas regras encontram-se descritas na Seção 2.3.

2.5 Transformada do Cosseno Discreto

Técnicas de esteganografia fundamentada em DCT se amparam na propriedade de que as imagens possuem certa redundância. Dessa forma, para cada componente de cor, a técnica utiliza a transformada de cosseno discreta para converter blocos sucessivos de 8×8 pixels em 64 coeficientes de uma DCT. Desta forma, os bits menos significativos dos coeficientes das DCT podem ser usados como bits redundantes para ocultar uma mensagem. Como as modificações realizadas na imagem estão concentradas no domínio da frequência, e não no domínio espacial, técnicas baseadas em DCT não deixam rastros perceptíveis para análises visuais [13]. Na implementação desse trabalho optou-se por esconder a mensagem secreta no componente DC da DCT, o qual é representado pelo coeficiente localizado no canto superior esquerdo da matriz de coeficientes DCT. Este coeficiente foi escolhido visando reduzir a distorção gerada ao armazenar a mensagem na imagem de capa, dado que ele representa o valor médio da cor de uma região 8x8 da imagem. Esta escolha foi feita com base em testes empíricos e dados existentes na literatura [14].

O algoritmo desenvolvido para esta técnica é muito semelhante ao algoritmo da técnica LSB, sendo que sua principal diferença ocorre antes de chamar a função $getBits(i_c)$ (linha 1), momento em que a imagem é convertida para o formato YCbCr e é aplicada a transformada do cosseno discreto na imagem, assim como após o procedimento de retorno (linha 13) é realizada a operação inversa. A transformada do cosseno discreto 2D é apresentada nas Equações (4), (5) e (6) [15]:

$$F(u,v) = C_u C_v \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2M}\right) \tag{4}$$

$$C_u = \begin{cases} \frac{1}{\sqrt{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1, 2, ..., N-1 \end{cases} \tag{5}$$

$$C_v = \begin{cases} \frac{1}{\sqrt{M}} & v = 0 \\ \sqrt{\frac{2}{M}} & v = 1, 2, ..., M-1 \end{cases} \tag{6}$$

onde $F(u,v)$ representa o valor da posição (u,v) na imagem após a transformada do cosseno discreto, $f(x,y)$ representa o valor da posição (x,y) na imagem antes da transformada do cosseno discreto. Note que C_u e C_v são constantes definidas nas Equações (5) e (6), onde N e M representam as dimensões da imagem, largura e altura, respectivamente.

As imagens digitais utilizadas neste trabalho são armazenadas no formato de matrizes, sendo que uma imagem RGB apresenta três matrizes, uma para representar cada canal. Como cada canal apresenta duas dimensões (largura e altura), é utilizada a transformada do cosseno discreto 2D.

2.6 Transformada Rápida de Fourier

A técnica de esteganografia baseada na transformada rápida de Fourier é muito semelhante à técnica baseada na transformada do cosseno discreto. Porém, no cálculo da transformada rápida de Fourier 2D utilizamos:

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y)\, exp\left(-j\frac{2\pi(ux+vy)}{N}\right), \tag{7}$$

onde $F(u,v)$ representa o valor da posição (u,v) na imagem após a transformada rápida de Fourier, $f(x,y)$ representa o valor da posição (x,y) na imagem antes da transformada rápida de Fourier. Como antes, (N,M) representam as dimensões da imagem, largura e altura, respectivamente.

Assim como na técnica DCT, a mensagem secreta é armazenada nos bits menos significativos do elemento localizado no canto superior esquerdo da matriz resultante da aplicação da transformada em cada bloco 8×8 da imagem. Portanto o algoritmo desenvolvido para esta técnica é muito semelhante ao pseudocódigo apresentado no

Algoritmo 1, para a técnica LSB, sendo que sua principal diferença é que antes de chamar a função $getBits(i_c)$ (linha 1) a imagem é convertida para o formato YCbCr e é aplicada a transformada rápida de Fourier na imagem, e após o retorno do procedimento (linha 13) é realizada a operação inversa.

Neste artigo, a técnica de esteganografia baseada em transformada rápida de Fourier foi implementada visando realizar uma comparação empírica dos resultados dos experimentos computacionais desta técnica e da técnica baseada em transformada do cosseno discreto. Ambas as técnicas pertencem ao conjunto de técnicas no domínio da frequência, enquanto as demais técnicas apresentadas atuam no domínio espacial.

2.7 Comparação das técnicas apresentadas

A Tabela 2 apresenta uma comparação das seis técnicas de esteganografia digital apresentadas neste artigo. As duas primeiras técnicas disponibilizam dois bits por pixel para a mensagem secreta, portanto permitem que sejam escondidas mensagens maiores em comparação com as demais técnicas. É importante ressaltar que as técnicas SSB-4 e SSB-N realizam uma normalização no valor de cada pixel visando reduzir a distorção gerada na imagem de capa e dificultar a detecção da esteganografia digital. Para recuperar a mensagem secreta utilizando a técnica SSB-N é necessário ter conhecimento da sequência dos índices que foram utilizados para esconder a mensagem, visto que esta sequência é gerada aleatoriamente.

Tabela 1: Resultados comparativos das técnicas em estudo.

	LSB RGB	LSB Escala de Cinza	SSB-4	SSB-N	DCT	FFT
Domínio	Espacial	Espacial	Espacial	Espacial	Frequência	Frequência
Bits por pixel utilizados para mensagem secreta	2	2	1	1	1	1
Utiliza normalização visando reduzir a distorção gerada			✓	✓		
Necessário conhecer algum dado adicional para revelar a mensagem				✓		
Utiliza transformada para esconder a mensagem					✓	✓

2.8 Métricas de Avaliação e Comparação

Neste trabalho foram utilizados como métrica de qualidade das imagens os valores de MSE, PSNR e SSIM para mensurar a diferença entre a imagem original e a imagem resultante, a qual contém a mensagem oculta. O valor de MSE pode ser obtido através da Equação (8).

$$\text{MSE} = \frac{1}{MN} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left(f(x,y) - \tilde{f}(x,y) \right)^2, \tag{8}$$

onde (N, M) representam as dimensões da imagem, $f(x, y)$ representa o valor na posição (x, y) na imagem original e $\tilde{f}(x, y)$ representa o valor na posição (x, y) na imagem com a mensagem escondida.

O valor de PSNR é calculado através de

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_i^2}{\text{MSE}} \right), \tag{9}$$

onde MSE representa o valor calculado pela Equação (8) e MAX_i^2 representa o maior valor para um pixel presente na imagem elevado ao quadrado, geralmente este valor é $255^2 = 65025$.

O valor de SSIM (*Structural Similarity*) é calculado através da Equação (10):

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{10}$$

onde μ_x e μ_y representam o valor médio de x e de y, respectivamente. σ_x^2 e σ_y^2 são a variância de x e y, nessa ordem, e σ_{xy} representa a covariância entre x e y. Os coeficientes c_1 e c_2 são definidos por $c_1 = (k_1 L)^2$ e $c_2 = (k_2 L)^2$, sendo $L = 2^{\text{bits por pixel}} - 1$, $k_1 = 0,01$ e $k_2 = 0,03$.

3 TRABALHOS RELACIONADOS

O uso de esteganografia digital vem sendo amplamente adotado para validação da originalidade de arquivos de áudio, vídeo, imagem, entre outros, visando dificultar o compartilhamento não autorizado destes arquivos. Além disso, a esteganografia digital pode ser utilizada por usuários mal intencionados para enviar ou obter informações de forma ilegal, sem que este ato seja detectado por *softwares* desenvolvidos especialmente para este propósito. Na literatura existem diversos trabalhos dedicados a técnicas de esteganografia digital. Uma breve revisão da literatura é apresentada nessa seção.

Alguns trabalhos camuflam dados em imagens utilizando sistemas caóticos híbridos e na transformada de Framelet, como em [16], onde os autores utilizam a criptografia juntamente com a esteganografia, visando proteger transmissões de imagens em redes sociais. O trabalho apresenta uma comparação dos resultados obtidos utilizando a técnica proposta e o sistema caótico híbrido 2D. Os resultados dos experimentos realizados mostram que a técnica proposta pode efetivamente resistir a ataques diferenciais, estatísticos, de ruído ou de perda de dados. Outro trabalho que também utiliza mapas caóticos, porém baseado na transformada inteira de Wavelet é apresentado em [17]. Os autores propõem uma técnica de esteganografia baseada em mapa logístico modificado para apresentar um espaço chave maior, o qual visa aumentar a segurança dos métodos de esteganografia baseados em mapas logísticos. A conclusão é obtida realizando uma análise de performance da técnica proposta utilizando PSNR. Em [18] é apresentada uma técnica de esteganografia baseada em combinações não lineares de mapas caóticos de 1D, onde a informação é armazenada no bit menos significativo. A performance é analisada utilizando as medidas PSNR e MSE.

Alguns trabalhos utilizam técnicas de esteganografia digital baseadas na transformada do cosseno discreto. Dentre estes, o trabalho [19] realiza um estudo na área de Teoria da Informação onde são abordadas algumas técnicas de esteganografia digital e alguns algoritmos de criptografia, os quais são classificados em simétricos, assimétricos e os que contém a propriedade do segredo perfeito. No trabalho [14], o armazenamento de informações sensíveis concernentes a um reator nuclear é modulado por esteganografia digital. Para isso, os autores utilizam a transformada do cosseno discreto e armazenam as informações no bit menos significativo. A performance do algoritmo é analisada utilizando as medidas de PSNR e MSE. Não foram observadas distorções nas imagens resultantes, e

os dados escondidos foram recuperados com exatidão. A diferença entre o resto da divisão por 3, dos dois coeficientes da transformada, utilizada em [20], visa incorporar dois bits da forma compactada da mensagem secreta. Tal técnica garante uma redução das alterações visuais na imagem resultante, e proporciona uma capacidade de armazenar mensagens que apresentam até 16, 7% do tamanho da imagem que contém a mensagem. O foco principal do trabalho são imagens no formato JPEG. É utilizado um *dataset* de teste composto por 5 mensagens secretas de tamanhos variando de 134 a 840 caracteres, e 3 imagens com qualidade distinta (baixa, média e alta). Os resultados foram validados utilizando as métricas MAE, MSE e SNR.

Visando minimizar os efeitos visuais dos algoritmos de esteganografia, [21] propõe um método para medir a distorção gerada na imagem pela alteração de cada bit. O intuito dos autores é possibilitar a escolha dos bits que causam menores distorções visuais na imagem, a fim de escolhê-los para esconder a mensagem secreta. O trabalho apresenta a utilização de *softwares* detectores de esteganografia para validar o resultado e demonstram que o método proposto é efetivo no caso em que a mensagem secreta é armazenada em posições fixas.

Existem trabalhos focados em melhorar algoritmos de criptografia para que a imagem resultante não aparente ter sido gerada de maneira aleatória. Nesse contexto, [22] apresenta um algoritmo de criptografia que produz imagens cifradas que possuem algum significado visual, ou seja, não aparentam ter sido concebidas aleatoriamente. Este algoritmo criptografa uma determinada imagem transformando-a em uma imagem com aspecto visual aleatório. Em uma segunda etapa, tal imagem, de aspecto visual aleatório, é modificada para se assemelhar a uma outra imagem que contém algum significado visual. Um método de criptografia de imagens que se assemelha a esteganografia é proposto em [23]. Os autores mostram que a imagem resultante do processo de criptografia não parece ter sido gerada aleatoriamente como geralmente ocorre, sendo bem próxima a uma imagem real com significado visual. O trabalho apresentado em [23] propõe uma melhoria para o método proposto em [22].

Alguns trabalhos estão interessados na detecção de esteganografia digital em imagens. A utilização de detecção de assinatura para identificar esteganografia digital em imagens foi proposto por [24]. O trabalho tem como objetivo principal detectar mensagens, em arquivos BMP ou JPG, em posições fixas como o cabeçalho do arquivo ou bit menos significativo da imagem. Testes foram realizados em 500 imagens extraídas do Google. Os resultados mostraram uma boa acurácia do método proposto. Em [25] um modelo matemático é proposto para predição de esteganografia digital em imagens utilizando a média ou a variância. Experimentos computacionais foram realizados, levando a conclusão que a variância se mostrou mais efetiva na predição de esteganografia digital, embora a média também tenha sido eficiente.

Comparações entre técnicas distintas de esteganografia digital são realizadas por alguns trabalhos. Uma comparação entre as técnicas LSB, DCT e DWT (*Discrete Wavelet Transform*) pode ser encontrada em [26]. Uma comparação detalhada entre as técnicas DCT e DWT é apresentada em [27].

Este trabalho realiza uma comparação entre as técnicas LSB, LSB Escala de Cinza, SSB-4, Permutação Pseudo-Aleatória para o Índice do Bit, DCT e FFT. Além de comparar tais técnicas, disponibilizamos um *dataset* no *Kaggle* contendo 17.950 imagens com mensagens escondidas armazenadas, nas quais foram utilizadas as técnicas em estudo neste trabalho. Além disso, uma biblioteca para a linguagem *Python* foi disponibilizada no repositório *PyPI - Python Package Index*, a qual permite realizar a ocultação e revelação de mensagens em imagens com as técnicas anteriormente mencionadas. É importante ressaltar que este *dataset* e a biblioteca poderão contribuir em trabalhos futuros na área de esteganálise.

4 EXPERIMENTOS COMPUTACIONAIS

As técnicas de esteganografia apresentadas previamente são avaliadas através de alguns experimentos computacionais. Esta seção compara tais técnicas e discute os resultados obtidos. Os testes foram realizados em uma máquina com a seguinte configuração: AMD Ryzen 717008c, 32GB RAM, 1 TB HD.

4.1 Dataset

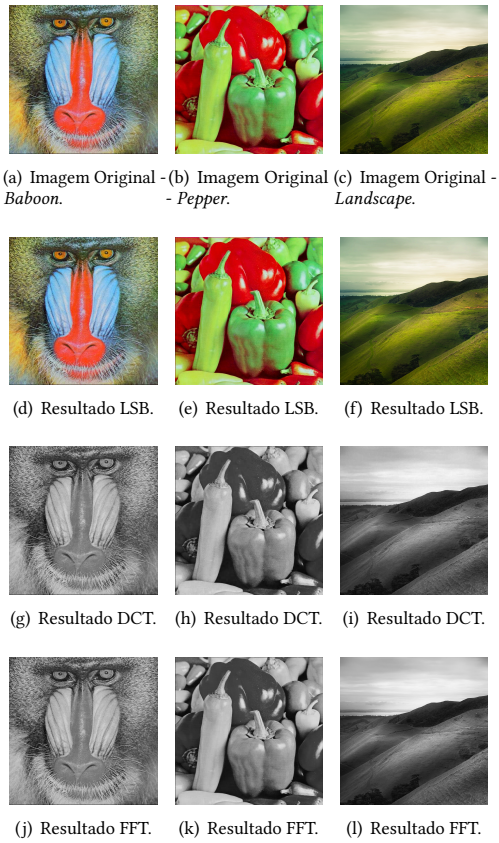
Foi gerado um *dataset* [3] de 18 GB de imagens com mensagens secretas escondidas utilizando as técnicas implementadas. Cabe ressaltar que tal conjunto de dados se encontra disponível para uso público.

4.2 Resultados

A Figura 5 apresenta as imagens resultantes das técnicas aplicadas em imagens em RGB e a Figura 6 exibe as imagens decorrente da aplicação das técnicas em estudo para imagens em escala de cinza. As imagens apresentadas nas Figuras 5(a), 5(b) e 5(c) são as imagens de capa (RGB). As Figuras 5(d)-5(l) mostram as imagens resultantes (com as mensagens escondidas) após a aplicação de cada técnica de esteganografia, para imagens em RGB, às respectivas imagens de capa. De forma geral pode-se observar que o resultado visual obtido apresenta pouca distorção visual, sendo que para o LSB é visualmente imperceptível, independente da imagem de capa utilizada. As imagens apresentadas nas Figuras 6(a), 6(b) e 6(c) são as imagens de capa para os testes em escala de cinza. Observando os resultados apresentados, através das imagens, para as técnicas em escala de cinza (Figuras 6(d)-6(j), 6(e)-6(k) e 6(f)-6(l)), podemos concluir que a comparação entre as imagens de capa e as imagens resultantes (com a mensagem escondida) foram satisfatórias, sendo impossível perceber visualmente a diferença entre as imagens de capa e as resultantes das técnicas aplicadas.

A Tabela 2 apresenta os valores das medidas de qualidade de MSE, PSNR E SSIM para as técnicas em estudo neste trabalho, referente às imagens de capa em comparação com a respectiva imagem pós-processada. Foi armazenado em cada uma delas um artigo retirado do *dataset* da Wikipedia. Nas imagens que estão sendo comparadas na tabela foi armazenado o mesmo artigo que foi escolhido aleatoriamente. Idealmente, para comparar os resultados, é necessário que todas as imagens mantenham a mesma proporção entre o tamanho da imagem de capa e tamanho da mensagem secreta armazenada. Visando manter esta proporção a mensagem secreta foi repetida n vezes até completar totalmente o espaço de armazenamento disponível para cada técnica em cada imagem de capa. É importante ressaltar que na técnica LSB, utilizando imagens RGB,

[3] www.kaggle.com/diegozanchett/digital-steganography



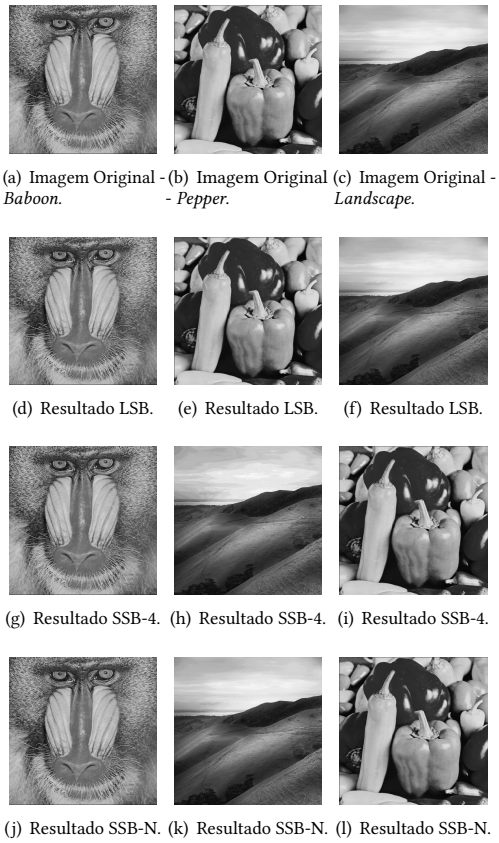
(a) Imagem Original - *Baboon.* (b) Imagem Original - *Pepper.* (c) Imagem Original - *Landscape.*

(d) Resultado LSB. (e) Resultado LSB. (f) Resultado LSB.

(g) Resultado DCT. (h) Resultado DCT. (i) Resultado DCT.

(j) Resultado FFT. (k) Resultado FFT. (l) Resultado FFT.

Figura 5: Resultado da aplicação das técnicas de esteganografia digital em três imagens em RGB.



(a) Imagem Original - *Baboon.* (b) Imagem Original - *Pepper.* (c) Imagem Original - *Landscape.*

(d) Resultado LSB. (e) Resultado LSB. (f) Resultado LSB.

(g) Resultado SSB-4. (h) Resultado SSB-4. (i) Resultado SSB-4.

(j) Resultado SSB-N. (k) Resultado SSB-N. (l) Resultado SSB-N.

Figura 6: Resultado da aplicação das técnicas de esteganografia digital em três imagens em escala de cinza.

é armazenada a mensagem secreta em todos os canais existentes nas imagens. Nas técnicas LSB, SSB-4, e "Permutação Pseudo Aleatória do Índice do Bit" utilizando imagens em escala de cinza, a mensagem secreta é armazenada em seu único canal. Portanto, a intensidade de cada canal nestas técnicas não gera uma mudança significativa na medida de qualidade. Para as técnicas DCT e FFT a medida de qualidade é sensível à intensidade do canal escolhido para armazenar a mensagem.

Tabela 2: Resultados das Métricas de Avaliação.

	Baboon			*Pepper*			*Landscape*		
Técnica	**MSE**	**PSNR (dB)**	**SSIM**	**MSE**	**PSNR (dB)**	**SSIM**	**MSE**	**PSNR (dB)**	**SSIM**
LSB	4.1721	41.9272	0.9972	4.1235	41.9780	0.9878	6.2399	40.1790	0.9824
LSB Escala de Cinza	0.0091	68.3163	0.9999	0.0090	68.0764	0.9999	2.3906	44.1388	0.9806
SSB-4 Escala de Cinza	0.0219	64.5051	0.9999	0.0244	63.7639	0.9999	12.0203	37.3316	0.9122
Permutação Pseudo Aleatória em Escala de Cinza	0.2501	53.9414	0.9999	0.1052	57.4182	0.9998	3.9873	42.0898	0.9539
DCT	5.9931	40.3542	0.9987	5.8939	40.4267	0.9961	5.9536	40.3829	0.9950
FFT	30.5323	33.28	0.9863	58.1736	30.4835	0.9270	103.9995	27.9604	0.8814

Os melhores valores para medidas de qualidade foram obtidos com as técnicas LSB em escala de cinza, SSB-4 em escala de cinza e permutação pseudo aleatória do índice do bit, devido ao fato de ambas modificarem principalmente os bits menos significativos do valor que representa a intensidade de cada pixel da imagem. As demais técnicas baseadas em escala de cinza apresentaram resultados razoáveis nas métricas obtidas a partir dos experimentos realizados.

A técnica DCT altera os bits menos significativos dos coeficientes encontrados ao calcular a transformada do cosseno discreto da imagem. Neste trabalho foram alterados apenas o coeficiente DC, evitando a perda de dados durante o processo de compressão e minimizando a distorção da imagem.

A técnica FFT é equivalente à técnica DCT, porém aqui utiliza-se a transformada discreta de Fourier. Observando os dados da Tabela 2 nota-se que esta técnica apresenta uma distorção maior do que a técnica DCT.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma comparação entre os métodos LSB, LSB Escala de Cinza, SSB-4 Escala de Cinza, SSB-N Escala de Cinza, DCT e FFT para utilização em esteganografia digital através de medidas de qualidade como MSE, PSNR e SSIM. O método SSB-N foi proposto neste trabalho visando garantir mais segurança no armazenamento das mensagens secretas e obter melhores resultados para as medidas de qualidade em comparação ao método SSB-4, no qual este método é baseado. É importante ressaltar que neste trabalho também disponibilizamos um *dataset* de esteganografia digital na plataforma *Kaggle* e uma biblioteca para linguagem *Python* na plataforma *PyPI*, que permite esconder e revelar mensagens secretas em imagens.

Em trabalhos futuros pretende-se propor algoritmos para correção dos erros ocorridos na recuperação das mensagens secretas utilizando as técnicas DCT e FFT, bem como realizar estudos utilizando algoritmos de esteganografia associados a algoritmos para correção de erros já existentes.

6 AGRADECIMENTOS

O presente trabalho foi realizado com apoio do programa de Iniciação Científica do CEFET/RJ - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - Rio de Janeiro – Brasil.

REFERÊNCIAS

[1] Eduardo Pagani Julio, Wagner Gaspar Brazil, and Célio Vinicius Neves Albuquerque. Esteganografia e suas aplicações. *Livro de Minicursos do SBSEG. Rio de Janeiro: Sociedade Brasileira de Computação*, 7:54–102, 2007.

[2] Diego Fiori de Carvalho, Rafael Chies, and Rudinei Goularte. Mp4stego: esteganografia em vídeos mpeg-4. In *Proceedings of the 14th Brazilian Symposium on Multimedia and the Web*, pages 154–161, 2008.

[3] M. Sharifzadeh, M. Aloraini, and D. Schonfeld. Adaptive batch size image merging steganography and quantized gaussian image steganography. *IEEE Transactions on Information Forensics and Security*, 15:867–879, 2020.

[4] Laura Cristina Machado Coelho and Ricardo Jorba Bento. Ferramentas de esteganografia e seu uso na infowar. *ICCyber'2004*, page 14, 2004.

[5] Abner da Silva Netto and Marco Antonio Pinheiro da Silveira. Gestão da segurança da informação: fatores que influenciam sua adoção em pequenas e médias empresas. *JISTEM-Journal of Information Systems and Technology Management*, 4 (3):375–397, 2007.

[6] Nursel Selver Ruzgar. A research on the purpose of internet usage and learning via internet. *Turkish Online Journal of Educational Technology-TOJET*, 4(4):27–32, 2005.

[7] Mehdi Hussain, Ainuddin Wahid Abdul Wahab, Yamani Idna Bin Idris, Anthony TS Ho, and Ki-Hyun Jung. Image steganography in spatial domain: A survey. *Signal Processing: Image Communication*, 65:46–66, 2018.

[8] Oleg O Evsutin, Anna S Melman, and Roman V Meshcheryakov. Digital steganography and watermarking for digital images: a review of current research directions. *IEEE Access*, 2020.

[9] Fabien AP Petitcolas, Ross J Anderson, and Markus G Kuhn. Information hiding-a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[10] Peter Wayner. *Disappearing Cryptography: Information Hiding: Steganography & Watermarking*. Morgan Kaufmann, 2002.

[11] Namita Tiwari and Dr Madhu Shandilya. Evaluation of various lsb based methods of image steganography on gif file format. *International Journal of Computer Applications*, 6(2):1–4, 2010.

[12] José Marconi Rodrigues, JR Rios, and William Puech. Ssb-4 system of steganography using bit 4. In *WIAMIS: Workshop on Image Analysis for Multimedia Interactive Services*, 2004.

[13] Niels Provos and Peter Honeyman. Hide and seek: An introduction to steganography. *IEEE security & privacy*, 1(3):32–44, 2003.

[14] Sahar A El_Rahman. A comparative analysis of image steganography based on dct algorithm and steganography tool to hide nuclear reactors confidential information. *Computers & Electrical Engineering*, 70:380–399, 2018.

[15] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[16] Y Khedmati, R Parvaz, and Y Behroo. 2d hybrid chaos map for image security transform based on framelet and cellular automata. *Information Sciences*, 512: 855–879, 2020.

[17] Milad Yousefi Valandar, Peyman Ayubi, and Milad Jafari Barani. A new transform domain steganography based on modified logistic chaotic map for color images. *Journal of Information Security and Applications*, 34:142–151, 2017.

[18] Sajjad Shaukat Jamal, Shabieh Farwa, Ali H Alkhaldi, Muhammad Aslam, and Mohammad Asif Gondal. A robust steganographic technique based on improved chaotic-range systems. *Chinese Journal of Physics*, 61:301–309, 2019.

[19] Fábio Borges de Oliveira. Análise da segurança de criptografia e esteganografia em seqüências de imagens. 2007.

[20] Abdelhamid Awad Attaby, Mona FM Mursi Ahmed, and Abdelwahab K Alsammak. Data hiding inside jpeg images with high resistance to steganalysis using a novel technique: Dct-m3. *Ain Shams Engineering Journal*, 9(4):1965–1974, 2018.

[21] Junhong Zhang, Wei Lu, Xiaolin Yin, Wanteng Liu, and Yuileong Yeung. Binary image steganography based on joint distortion measurement. *Journal of Visual Communication and Image Representation*, 58:600–605, 2019.

[22] Long Bao and Yicong Zhou. Image encryption: Generating visually meaningful encrypted images. *Information Sciences*, 324:197–207, 2015.

[23] Ali Kanso and Mohammad Ghebleh. An algorithm for encryption of secret images into meaningful images. *Optics and lasers in engineering*, 90:196–208, 2017.

[24] Pengjie Cao, Xiaolei He, Xianfeng Zhao, and Jimin Zhang. Approaches to obtaining fingerprints of steganography tools which embed message in fixed positions. *Forensic Science International: Reports*, 1:100019, 2019.

[25] Natasha N Paiva, Renato Portugal, Pedro Carlos S Lara, Centro Federal de Educaç ao Tecnológica, and Celso Suckow da Fonseca-CEFET. Prediction of steganography in images using a mathematical model. *XXXVI Ibero-Latin American Congress on Computational Methods in Engineering*, 2015.

[26] G Stuti, R Arun, and K Manpreet. A review of comparison techniques of image steganography. *IOSR Journal of Electrical and Electronics Engineering*, 6(1):41–48, 2013.

[27] S Saejung, A Boondee, J Preechasuk, and C Chantrapornchai. On the comparison of digital image steganography algorithm based on dct and wavelet. In *2013 International Computer Science and Engineering Conference (ICSEC)*, pages 328–333. IEEE, 2013.