CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA — CEFET/RJ *CAMPUS* PETRÓPOLIS
CURSO: BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Um Estudo Sobre Aplicação de Redes Neurais Convolucionais e Técnicas de Codificação de Canal em Esteganografia Digital

Diego Henrique Baltar Zanchett

PETRÓPOLIS
2022

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA — CEFET/RJ *CAMPUS* PETRÓPOLIS
CURSO: BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Um Estudo Sobre Aplicação de Redes Neurais Convolucionais e Técnicas de Codificação de Canal em Esteganografia Digital

Diego Henrique Baltar Zanchett

Trabalho de Conclusão de Curso apresentado ao CEFET/RJ — *campus* Petrópolis, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Orientadora: Profa. Laura Silva de Assis
Co-Orientador: Prof. Diego Barreto Haddad

PETRÓPOLIS
2022

Cefet/RJ – Sistema de Bibliotecas / Uned Petrópolis

Z27 Zanchett, Diego Henrique Baltar

Um Estudo Sobre Aplicação de Redes Neurais Convolucionais e Técnicas de Codificação de Canal em Esteganografia Digital / Diego Henrique Baltar Zanchett. - 2022.
73f.

Trabalho de conclusão de curso (Curso de Bacharelado em Engenharia de Computação) – Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Petrópolis (RJ), 2022.

Bibliografia: p. 66 - 73.

Orientadora: Profª. Laura Silva de Assis
Co-Orientador: Prof. Diego Barreto Haddad

1. Internet (Redes de computação) - Medidas de segurança. 2. Processamento de Imagens. 3. Esteganografia digital - Técnicas. 4. Redes Neurais. 5. Autoencoders. 6. Monografia. I. Assis, Laura Silva de.(Orient.). II. Haddad, Diego B. (Coorient.). II. Título.

CDD 005.8

Elaborada por Luciana de Souza Castro CRB7/ 4812

CEFET/RJ
campus Petrópolis

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA — CEFET/RJ *CAMPUS* PETRÓPOLIS
CURSO: BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Um Estudo Sobre Aplicação de Redes Neurais Convolucionais e Técnicas de Codificação de Canal em Esteganografia Digital

Diego Henrique Baltar Zanchett

Trabalho de Conclusão de Curso apresentado ao CEFET/RJ — *campus* Petrópolis, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Orientadora: Profa. Laura Silva de Assis
Co-Orientador: Prof. Diego Barreto Haddad

Aprovado por:

Prof. Laura Silva de Assis, D.Sc. (Orientadora)

Prof. Diego Barreto Haddad, D.Sc. (Co-Orientador)

Prof. Luis Domingues Tome Jardim Tarrataca, Ph.D.

Prof. Pedro Carlos da Silva Lara, D.Sc.

Março de 2022

DEDICATÓRIA

Dedico este trabalho aos meus pais, Rita de Cássia da Silva Baltar e Nilceo Pedro Zanchett, que sempre me apoiaram e me incentivaram a estudar durante toda minha trajetória.

Dedico este trabalho à minha orientadora e ao meu co-orientador, por dedicar uma parte de seu tempo para guiar este projeto de pesquisa.

Dedico este trabalho aos meus orientadores nos projetos de iniciação científica pelo apoio técnico e por me incentivaram a ler artigos, participar de congressos, promovendo uma experiência de muito aprendizado.

Dedico aos professores do curso de Engenharia de Computação da instituição CEFET/RJ, onde obtive grande parte do conhecimento necessário para realização deste trabalho.

AGRADECIMENTO

Agradeço à minha orientadora, a professora D.Sc. Laura Silva de Assis e meu co-orientador professor D.Sc. Diego Barreto Haddad por ter aceitado me acompanhar neste projeto, fornecendo questionamentos e indicações relevantes que permitiram desenvolver este trabalho.

Agradeço ao professor M.Sc. Jurair Rosa de Paula Junior que junto com os professores orientador e co-orientador deste trabalho me apoiou em projetos de iniciação cientifica durante minha trajetória acadêmica.

Agradeço a todos os professores do curso de engenharia de Computação pelo conhecimento compartilhado. Agradeço também à instituição CEFET/RJ por fornecer os recursos físicos e tecnológicos necessários para realização deste trabalho.

RESUMO

Este trabalho apresenta um estudo da aplicação de redes neurais convolucionais em esteganografia digital. Esteganografia digital consiste em algumas técnicas que permitem armazenar ocultamente um arquivo digital dentro de outro arquivo digital. Neste estudo são comparadas diversas topologias e hiper-parâmetros para uma rede neural convolucional, visando avaliar quais obterão os melhores resultados nesta tarefa. É apresentada uma técnica inédita de pré-processamento e pós-processamento que utiliza ruído e autoencoders para ocultar e extrair arquivos em imagens, utilizando uma taxa para inserção da mensagem oculta de 0.33 bit por pixel. Como no processo de ocultação e extração da mensagem secreta utilizando autoencoders podem surgir erros no arquivo secreto, é avaliada a utilização de uma técnica de codificação de canal para corrigir estes erros. Além da técnica que utiliza redes neurais, são apresentadas e comparadas neste trabalho outras cinco técnicas para esteganografia digital. Sendo importante ressaltar que uma delas, a técnica SSB-N, é proposta neste trabalho. Foram realizados experimentos para identificar o comportamento de cada técnica de acordo com cinco propriedades: *i)* qualidade visual da imagem pós-processada, *ii)* local onde o conteúdo oculto é armazenado na imagem, *iii)* capacidade de armazenamento da técnica, *iv)* segurança da técnica e *v)* tempo de processamento. Dois *datasets* extraídos da *internet*, contendo 282Mb de imagens, foram utilizados nos experimentos realizados. Após a apresentação dos resultados numéricos é realizada uma análise dos valores obtidos nos experimentos, visando identificar qual técnica obteve o melhor desempenho para cada propriedade analisada.

Palavras-chave: Esteganografia Digital, Processamento de Imagens, Redes Neurais Convolucionais, Autoencoders, Segurança da Informação.

ABSTRACT

This work presents a study of the application of convolutional neural networks in digital steganography. Digital steganography consists of some techniques that allow you to hide a digital file inside another digital file. In this study, several topologies and hyperparameters for a convolutional neural network are compared, to evaluate which ones will obtain the best results in this task. An unprecedented pre-processing and post-processing technique is presented that uses noise and autoencoders to hide and extract files in images, using a hidden message insertion rate of 0.33 bit per pixel. As in the process of hiding and extracting the secret message using autoencoders, errors may appear in the secret file, the use of a channel coding technique to correct these errors is evaluated. In addition to the technique that uses neural networks, five other techniques for digital steganography are presented and compared in this work. It is important to emphasize that one of them, the SSB-N technique, is proposed in this work. Experiments were carried out to identify the behavior of each technique according to five properties: *i)* visual quality of the post-processed image, *ii)* where the hidden content is stored in the image, *iii)* technique storage capacity, *iv)* technique security and *v)* processing time. Two datasets extracted from internet, containing 282Mb of images, were used in the experiments. After presenting the numerical results, an analysis of the values obtained in the experiments is performed, aiming to identify which technique obtained the best performance for each analyzed property.

Key-words: Digital Steganography, Image Processing, Convolutional Neural Networks, Autoencoders, Information Security.

LISTA DE FIGURAS

LISTA DE TABELAS

Lista de Siglas

AES	-	Advanced Encryption Standard
AI	-	Artificial Intelligence
AIC	-	Advanced Image Coding
ASCII	-	American Standard Code for Information Interchange
BMP	-	Bitmap
CD	-	Compact Disc
CIE	-	Commission Internationale de l'Éclairage
CNN	-	Convolutional Neural Network
CPU	-	Central Processing Unit
DCT	-	Discrete Cosine Transform
DES	-	Data Encryption Standard
DICOM	-	Digital Imaging and Communications in Medicine
DPCM	-	Differential pulse-code modulation
DSS	-	Digital Signature Standard
DVD	-	Digital Versatile Disc
ECC	-	Elliptical Curve Cryptography
FFT	-	Fast Fourier Transform
FITS	-	Flexible Image Transport System
GAN	-	Generative Adversarial Network
GB	-	Gigabyte
GHz	-	Gigahertz
GIF	-	Graphics Interchange Format
GPU	-	Graphics Processing Unit
HD	-	Hard Disk
HSL	-	Hue, Saturation, and Lightness
HSV	-	Hue, Saturation, and Value
IA	-	Inteligência Artificial
IDEA	-	International Data Encryption Algorithm
IoT	-	Internet of Things
ISO	-	International Organization for Standardization
ITU	-	International Telecommunication Union
JFIF	-	JPEG File Interchange Format
JPEG	-	Joint Photographics Experts Group
LPIPS	-	Learned Perceptual Image Patch Similarity
LSB	-	Least Significant Bit
MCU	-	Minimum Coded Unit
MSE	-	Mean Squared Error
NASA	-	National Aeronautics and Space Administration
OCR	-	Optical Character Recognition
PCIe	-	Peripheral Component Interconnect Express
PNG	-	Portable Network Graphics
PSNR	-	Peak Signal-to-Noise Ratio
QR	-	Quick Response

RAM	-	Random-Access Memory
RGB	-	Red, Green, and Blue
RGBA	-	Red, Green, Blue, and Alpha
RLE	-	Run-Length Encoding
RNA	-	Rede Neural Artificial
RSA	-	Rivest Shamir Adleman
SSB-4	-	System of Steganography using bit 4
SSB-N	-	System of Steganography using bit N
SSIM	-	Structural Similarity Index
SVG	-	Scalable Vector Graphics
TIFF	-	Tagged Image File Format
TXT	-	Text Format
VGG	-	Visual Geometry Group
VM	-	Virtual Machines
XML	-	eXtensible Markup Language

SUMÁRIO

1 Introdução

A popularização e crescimento da *internet* proporcionou o surgimento de diversos novos serviços, empresas e soluções (1). A partir deste crescimento acelerado surgiram diversas melhorias em diferentes setores, todavia também surgiram novos problemas. Dentre estes problemas, podemos citar: as distintas violações de privacidade (2), grandes vazamentos de dados (3), propagação de notícias falsas (4), entre outros. A partir do aumento do poder computacional e do desenvolvimento de sistemas de inteligência artificial como *deep learning* surgiram as *deep fakes*, que consistem em manipulações de vídeos, fotos ou áudios visando enganar um público alvo através de alterações neste formato de conteúdo para que uma pessoa relevante pareça estar comentando sobre uma informação falsa. Estes conteúdos falsos são comumente gerados por sistemas de *deep learning*, treinados para esta finalidade (5).

Atualmente, uma grande quantidade de dados é gerada diariamente e disponibilizada na *internet*. Observando os dados de alguns serviços populares, em 2021, a cada minuto 197,7 milhões de e-mails foram enviados. Estes dados são compostos por diversas categorias de informações que comumente são sensíveis e precisam de tratamentos adicionais para evitar que vazamentos e violações de privacidade ocorram (6). Um meio de proteger as informações que trafegam na *internet* consiste em utilizar criptografia para codificar as informações enquanto estão em trânsito na rede ou em repouso nos locais de armazenamento de dados.

Podemos classificar as duas categorias de técnicas de criptografia mais comuns em criptografia em: *i)* chave simétrica; e *ii)* chave assimétrica (7). A criptografia de chave simétrica utiliza a mesma chave para codificar e decodificar a mensagem. Por este motivo, tal técnica é utilizada principalmente em sistemas fechados. Já a criptografia de chave assimétrica utiliza duas chaves: uma chave pública para codificar a mensagem e uma chave privada para decodificá-la. É importante ressaltar que a chave privada deve ser mantida em segredo, entretanto a chave pública pode ser compartilhada. Esta técnica é utilizada para comunicação pela *internet*, já que é possível compartilhar a chave pública pela rede (7).

A criptografia utiliza uma chave criptográfica para codificar a mensagem a ser transmitida ocultando o conteúdo da mensagem, mas não a sua existência. Portanto, mesmo utilizando criptografia ainda é possível detectar que uma informação codificada está trafegando na rede. Visando dar ainda mais segurança à transmissão e ao armazenamento de informações sensíveis, é interessante ocultar a existência dessa informação. A esteganografia é uma técnica que visa ocultar a existência de um dado. Para isto, o dado a ser escondido é armazenado em redundâncias de uma informação pública (8).

A palavra esteganografia deriva dos vocábulos grego *steganós* (oculto) e *graphia* (escrita), significando portanto tratar-se de uma "escrita oculta". A primeira aparição de técnicas de esteganografia na história remete ao ano de 440 AC, quando Heródoto descreve em seu livro "Histórias" o momento em que Demeratus (personagem principal) escreve em uma placa de madeira uma mensagem informando sobre um ataque à Grécia. Após concluir a escrita, a placa é coberta com cera, visando ocultar a existência da mensagem (9).

Um registro mais recente da utilização de esteganografia ocorreu na Segunda Guerra Mundial, quando os alemães usaram pequenos pontos impressos que quando ampliados continham conteúdo de páginas datilografadas em sua dimensão normal (9). Atualmente a esteganografia é muito explorada em meios digitais para diversas finalidades. Entre elas podemos citar: *i)* aumentar a segurança na transmissão e armazenamento de informações sensíveis, *ii)* validação de autenticidade de arquivos, *iii)* rastreabilidade no compartilhamento de arquivos, *iv)* validação se determinado arquivo não foi editado, entre outras. (10).

Para validação de autenticidade e rastreabilidade de arquivos é geralmente armazenado, utilizando esteganografia digital, um selo no arquivo a ser compartilhado (10). A edição de arquivos nos quais existem dados ocultos pode afetar de modo irreversível a recuperação da informação codificada. Técnicas de esteganografia nas quais tais perdas ocorrem não apresentam a propriedade conhecida como "resistência à edição". A ausência desta propriedade pode ser útil quando empregada para aferir se um determinado arquivo foi ou não editado (11, 12). As categorias de arquivos digitais mais explorados para utilização em sistemas de esteganografia são as que apresentam mais redundância, como, por exemplo, fotos, vídeos e áudio. Arquivos digitais redundantes são aqueles que contém repetição da mesma informação em sua composição, por exemplo, ao representar uma imagem no formato matricial sem compressão os mesmos valores de cor serão repetidos para todos os pixels que contenham a respectiva cor. Contudo, também existem trabalhos que exploram a utilização de outros meios para aplicação de esteganografia digital, como textos e jogos (13).

Este trabalho aplica redes neurais convolucionais em esteganografia. Redes neurais convolucionais são uma categoria de rede neural artificial utilizada principalmente no processamento de imagens e vídeos. Redes neurais artificiais consistem em algoritmos de aprendizado de máquina inspirados na operação de neurônios no cérebro humano. É importante ressaltar que as redes neurais artificiais tradicionais não são otimizadas para o processamento de imagens, entretanto redes neurais convolucionais são otimizadas para o processamentos de imagens e tarefas de visão computacional (14) dado que a implementação de seus "neurônios" são inspirados no "lobo frontal", área responsável pelo processamento de estímulos visuais em humanos e outros animais (15).

O objetivo do presente trabalho é avaliar a utilização de redes neurais convolucionais na tarefa de ocultar uma sequência de *bits*, que representa um arquivo binário, dentro de uma imagem no formato PNG. Para isto será utilizado a biblioteca de aprendizado de máquina *Tensor Flow* e técnicas de codificação de canal para detectar e corrigir os *bits* da mensagem secreta que apresentarem algum erro após a recuperação da mesma.

1.1 Tema

O trabalho apresenta uma avaliação da aplicação de redes neurais convolucionais na tarefa de armazenar arquivos digitais nas partes redundantes de imagens, os quais são armazenados utilizando representação matricial. Para realizar tal tarefa é feito um estudo da aplicação de algumas técnicas de

aprendizado de máquina. Dado que o processo de recuperação do arquivo binário a partir da imagem utilizando redes neurais convolucionais pode ocasionar alguns erros no arquivo armazenado, serão então utilizadas técnicas de codificação de canal para correção destes erros.

1.2 Delimitação

O objetivo deste trabalho é estudar a aplicação de técnicas de aprendizado de máquina e codificação de canal em esteganografia digital para armazenar arquivos binários em arquivos de imagens. Não faz parte do escopo deste trabalho estudar o armazenamento de arquivos binários dentro de outras categorias de arquivos como textos, áudio, vídeos, entre outros.

Ao realizar a tarefa de esteganografia em imagens utilizando redes neurais é gerada alguma distorção nas imagens e alguns erros nos arquivos binários. O desenvolvimento deste trabalho visa minimizar tal distorção e os erros gerados durante o processamento realizado pela rede neural convolucional (CNN, do inglês *convolutional neural network*). Após a CNN terminar seu processamento, são utilizadas técnicas de codificação de canal para eliminar os erros encontrados nos arquivos binários. Para mensurar a distorção gerada nas imagens serão utilizadas métricas como erro absoluto médio, relação sinal ruído de pico, índice de similaridade estrutural, assim como métricas perceptuais.

1.3 Justificativa

O avanço tecnológico proporciona a conexão de diversos dispositivos à *internet*. Consequentemente, o volume de dados trafegado na rede está em constante crescimento. A partir do surgimento da Internet das Coisas (IoT, do inglês, *Internet of Things*), os dispositivos que se conectam à *internet* se tornaram cada vez mais diversificados. Isso permitiu o desenvolvimento de novas soluções que geram maior praticidade para as pessoas. Por exemplo, utilizando um dispositivo IoT é possível ligar o ar-condicionado no caminho de casa para que ao chegar em casa o ambiente esteja em uma temperatura agradável (16, 17).

Infelizmente, o aumento do tráfego de informações na rede também levou ao aumento do acesso não autorizado a dados sensíveis e à dificuldade no controle da privacidade na rede. Isto torna o desenvolvimento de soluções para proteção destes dados cada vez mais necessária (3). Algumas das áreas de estudo relacionadas a proteção de dados encontram-se a criptografia e a esteganografia.

A criptografia visa codificar uma mensagem para que ao ser interceptada por algum usuário não autorizado seja difícil à sua interpretação. Já a maioria dos métodos de esteganografia visam esconder a mensagem a ser protegida dentro de outra informação já disponível publicamente, visando confundir o interceptador para ser difícil a percepção da existência da mensagem escondida (18).

Uma parte dos métodos de esteganografia digital se baseia na propriedade que algumas categorias de informações digitais contém, que é a redundância, portanto é possível armazenar a mensagem secreta neste espaço redundante sem perder o significado da informação original. Porém, mesmo que seja possível manter o significado do arquivo original alguma distorção é gerada (19).

Entretanto, existem alguns métodos de esteganografia que se baseiam em gerar uma imagem que contenha um significado visual para os seres humanos enquanto represente a mensagem secreta para o algoritmo de extração da mensagem. Estas técnicas podem ser implementadas utilizando redes neurais adversárias generativas (GAN, do inglês, *Generative Adversarial Networks*), síntese de texturas, dentre outras técnicas (20).

1.4 Objetivos

A partir dos recentes avanços nos estudos de redes neurais, percebeu-se que tais técnicas são eficientes para realizar esteganálise, ou seja, detectar esteganografia em arquivos redundantes (21). Também notou-se que uma forma de evitar que a esteganografia seja detectada por estas redes neurais consiste em utilizar GANs para gerar uma imagem já contendo a mensagem secreta (20). Utilizar imagens sintéticas originadas a partir de redes neurais impede que o conjunto de imagens originais, ou seja, sem tratamento de esteganografia, seja encontrado. Com isto o treinamento das redes de esteganálise neste conjunto de imagens é inviabilizado. Como descrito na literatura, caso o treinamento das redes de esteganálise no conjunto de dados original seja inviabilizado a acurácia das redes de esteganálise tende a se reduzir (20). É importante ressaltar que uma das diferenças das técnicas de esteganografia baseadas em GANs em relação as demais, consiste no fato de que as técnicas baseadas em GANs não permitem que o usuário escolha a imagem redundante que será utilizada.

O objetivo deste trabalho é realizar um estudo utilizando CNNs com topologia correspondente a um *autoencoder* na tarefa de esteganografia digital. Esta topologia permite ao usuário escolher as imagens que serão utilizadas. Contudo, ao esconder a mensagem secreta, algumas partes da imagem serão re-geradas pelo autoencoder, ou seja, as técnicas de esteganografia baseadas em esconder a mensagem secreta em uma imagem pré-existente é mesclada com as técnicas baseadas em geração de imagens.

Portanto, deseja-se desenvolver uma técnica segura em relação às redes de esteganálise. Assim, as redes de esteganálise devem apresentar baixa acurácia ao tentar detectar a esteganografia. Deste modo, espera-se que a imagem pós-processada apresente pouca distorção. É importante ressaltar que em nossa abordagem é possível que o usuário escolha a imagem a ser processada para esconder a mensagem.

1.5 Metodologia

A metodologia proposta nesse trabalho é baseada nos conceitos e técnicas supracitadas. São utilizadas duas redes neurais convolucionais como autoencoders visando a redução do nível de ruído. A mensagem secreta a ser escondida é composta por um arquivo binário genérico. Nos experimentos realizados neste trabalho foram gerados arquivos no formato `txt` contendo uma *string* aleatória. Inicialmente, o arquivo binário será pré-processado, e alguns *bits* redundantes são adicionados, os quais são provenientes de alguma técnica de codificação de canal. Em seguida, esta sequência de *bits* do

arquivo binário, após o pré-processamento, é adicionada à imagem que será processada na forma de ruído.

A imagem contendo o ruído que representa o arquivo secreto será processada por uma CNN com uma topologia de um *denoising autoencoder*. Esta rede é responsável por esconder este ruído na imagem e consequentemente esconder os dados que estão sendo armazenados. Para revelar o conteúdo secreto, a imagem é processada por outra CNN que terá a função de evidenciar o ruído que representa a mensagem secreta, e em seguida, esta imagem processada pela CNN será tratada visando converter o ruído em uma sequência de *bits*. Posteriormente estes *bits* serão processados pelas técnicas de codificação de canal para detecção e correção de erros e convertidos novamente para um arquivo binário. É importante ressaltar que as CNNs que serão responsáveis por esconder o ruído na imagem e revelar tal ruído serão treinadas paralelamente. Desta forma, o aprendizado, tanto no processo de esconder, quanto no de revelar o ruído, ocorre simultaneamente.

1.6 Descrição

O restante deste trabalho está organizado em mais sete capítulos. O Capítulo 2 apresenta a fundamentação teórica referente aos principais conceitos utilizados neste trabalho. O Capítulo 3 aborda redes neurais. Os principais conceitos de esteganografia e esteganálise utilizados neste trabalho são introduzidos no Capítulo 4. No Capítulo 5 são descritos os principais trabalhos relacionados. O Capítulo 6 apresenta a solução proposta neste trabalho. O Capítulo 7 apresenta e discorre sobre os resultados alcançados. As considerações finais são realizadas no Capítulo 8.

2 Fundamentação Teórica

Neste capítulo são abordadas os principais conceitos e técnicas utilizados no desenvolvimento deste trabalho. Assim, são apresentados: *i)* os formatos para representação de imagens digitais, vetorial e matricial; *ii)* os espaços de cores nas quais as imagens digitais podem ser representadas: *a)* escala de Cinza, *b)* RGB, *c)* RGBA, *d)* YCbCr, *e)* CMYK, *f)* HSV, *g)* HSL, *h)* CIE-Lab; *iii)* os formatos de arquivos que podem ser utilizados: *a)* svg, *b)* bmp, *c)* jpeg, *d)* png, *e)* gif, entre outros. Também são introduzidas técnicas de criptografia para proteção de dados armazenados e transmitidos, técnicas de codificação de canal para detectar e corrigir erros ao armazenar ou transmitir dados (os códigos de *Hamming* e de *Reed Solomon*). Assim como, medidas de semelhanças entre imagens como erro quadrático médio (MSE), relação sinal ruído de pico (PSNR), similaridade estrutural (SSIM) e métricas perceptuais.

2.1 Representação de Imagens Digitais

Em um meio digital as imagens podem ser representadas de duas formas: *1)* através da representação vetorial; ou *2)* utilizando a representação matricial. A seguir é apresentada cada categoria de representação, assim como seus pontos positivos e negativos.

2.1.1 Representação Vetorial

A representação vetorial de uma imagem é realizada por vetores matemáticos que reproduzem cada forma da imagem. Ou seja, caso a imagem contenha um círculo, será armazenado um vetor contendo os parâmetros necessários para reproduzir determinado círculo (22). A especificação de cada formato de imagem vetorial determina como estes vetores são organizados e armazenados. Por exemplo, um formato pode definir o vetor do círculo a ser armazenado como mostrado na Equação (1):

$$\overrightarrow{v} = (x, y, r, c), \tag{1}$$

onde $\overrightarrow{v}$ retrata o vetor a ser armazenado, (x, y) são a posição do centro do círculo, r o raio e c é um valor inteiro que indica a cor (23). Neste formato de representação os demais pontos que caracterizam o círculo e deverão ser apresentados na cor c podem ser encontrados através da expressão matemática mostrada na Equação (2):

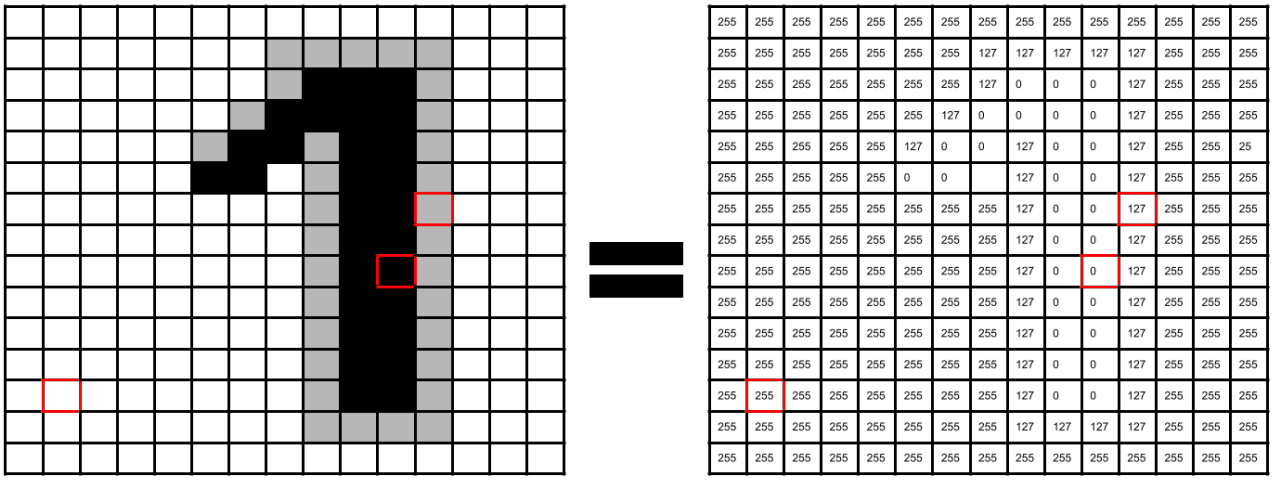
$$x^2 + y^2 \leq r^2. \tag{2}$$

As principais vantagens da representação vetorial consistem em não ocupar muito espaço de armazenamento, pois não são armazenadas informações redundantes e não ocorre perda de qualidade ao ampliar ou reduzir a imagem. Este formato é utilizado principalmente em projetos multimídias que requerem impressões em diferentes dimensões (23).

2.1.2 Representação Matricial

A representação matricial de uma imagem é composta por uma ou mais matrizes nas quais cada valor de cada matriz representa a intensidade de um pixel da imagem e determinado canal. A quantidade de matrizes/canais e o intervalo de valores utilizados é especificado na definição de cada formato de arquivo (24). Nos formatos mais populares são utilizados uma, três ou quatro matrizes com os valores variando no intervalo entre 0 e 255 (25).

Figura 1: Representação matricial de uma imagem.



Fonte: Próprio autor

A principal vantagem da representação matricial consiste em resolver um dos problemas da representação vetorial, que é a dificuldade de encontrar os vetores necessários para representar uma imagem capturada a partir de uma câmera ou um escâner. A representação matricial é utilizada principalmente em fotografias e vídeos (25).

Esta forma de registrar imagens leva ao armazenamento de muitas informações redundantes. Por exemplo, dado que determinada região da imagem seja composta por uma cor sólida é necessário repetir o valor que representa esta cor desta região para todos os pixels que a compõe (25). Por consequência, esta característica da representação matricial favorece a aplicação de técnicas de esteganografia digital, já que é possível utilizar esta informação redundante para armazenar a mensagem secreta (26).

2.1.3 Modelo de Cores

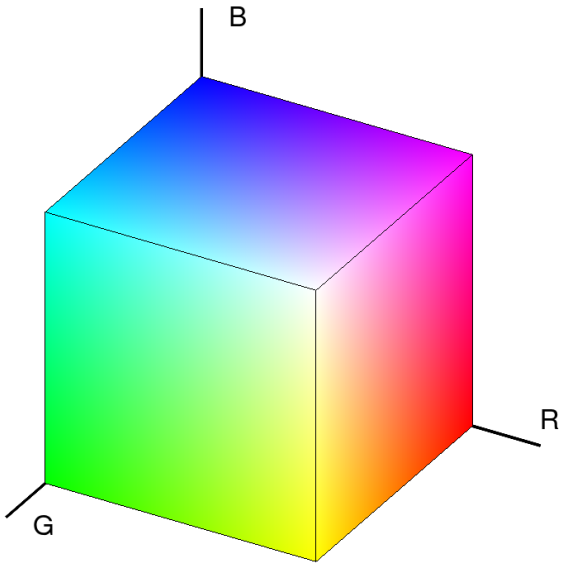
No momento em que uma pessoa visualiza algum objeto, é possível identificar diversas características do objeto. Algumas destas características podem ser reconhecidas através da forma do objeto, da sombra projetada por ele, da reflexão de outros objetos, bem como de sua cor. Portanto, ao compor uma cena através de uma imagem digital é importante representar cores visando permitir

o máximo de proximidade com a cena original. Para que um computador possa armazenar cores e o monitor consiga representá-las é necessário a definição de um modelo de cores.

Um modelo de cores pode ser definido como a maneira que um ser humano consegue visualizar cores de acordo com alguns atributos, como, por exemplo, a luminância (27). Um modelo de cores consiste em uma representação matemática que permite definir cores por valores numéricos, e existem modelos ideais para cada categoria de aplicação. Os principais modelos de cores são: RGB, RGBA, CMY, CMYK, YCbCr e HSI.

O modelo de cor RGB (Vermelho, Verde e Azul, do inglês, *Red*, *Green*, e *Blue*) representa uma cor qualquer através da intensidade das três cores primárias aditivas (amarelo, vermelho e azul). Neste modelo, quando as três cores primárias estão representadas em sua intensidade máxima, obtém-se branco. O modelo de cor RGB é muito utilizado em aplicações de computação gráfica. Adicionando o canal *alpha* ao modelo de cor RGB se obtém o modelo de cor RGBA, onde o canal *alpha* é responsável por armazenar a transparência de determinado pixel da imagem (28). Este modelo é reproduzido na Figura 2.

Figura 2: Espaço de Cores RGB.



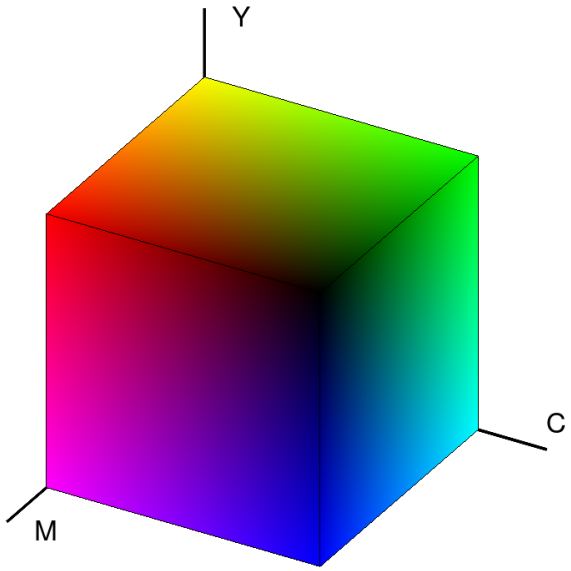
Fonte: Próprio autor

O modelo de cor CMY representa uma cor qualquer baseado na intensidade de três cores complementares subtrativas (Ciano, Magenta e Amarelo). Neste modelo, quando as três cores complementares estão representadas em sua intensidade máxima, obtém-se preto. Este modelo é muito utilizado em dispositivos de impressão. Adicionando-se um canal para representar a intensidade da cor preta obtém se o modelo de cor CMYK (28). A conversão de RGB para CMYK pode ser realizada através das Equações (3) e (4).

$$\begin{aligned} Ciano &= 1 - R, \\ Magenta &= 1 - G, \\ Amarelo &= 1 - B. \end{aligned} \tag{3}$$

$$Preto = \min(Ciano,\ Magenta,\ Amarelo). \tag{4}$$

Figura 3: Espaço de Cores CYMK.



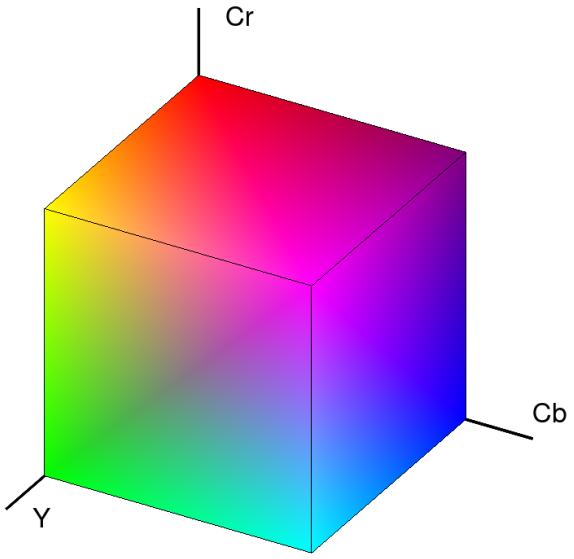
Fonte: Próprio autor

O modelo de cor YCbCr é composto por três componentes: *i)* a luminância, *ii)* a crominância azul, e *iii)* a crominância vermelha. Este formato é muito utilizado na transmissão de vídeo digital, pois permite uma compressão mais eficiente. Dado que o ser humano tem mais facilidade para visualizar alterações na luminância que na crominância, a perda de algumas informações na crominância durante a compressão não gera impacto significativo (28). A conversão de RGB para YCbCr pode ser realizada através da Equação (5):

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B, \\ Cb &= -0.169R + -0.331G + 0.500B, \\ Cr &= 0.500R + -0.419G + -0.081B. \end{aligned} \tag{5}$$

O modelo de cor HSI é baseado no sistema visual humano. Este modelo utiliza coordenadas cilíndricas para composição das cores RGB. O H expressa a matiz, do inglês *hue*, que representa a pureza da cor. O S significa a saturação, do inglês *saturation*, que representa o grau da cor branca

Figura 4: Espaço de Cores YCbCr.



Fonte: Próprio autor

incorporada na cor em questão. O I significa intensidade, do inglês *intensity*, que representa a intensidade da cor. Este modelo de cor é utilizado em algumas aplicações de edição de imagens e visão computacional (28). A conversão de RGB para HSI pode ser realizada através das Equações (6), (7) e (8):

$$I = \frac{R+G+B}{3}, \tag{6}$$

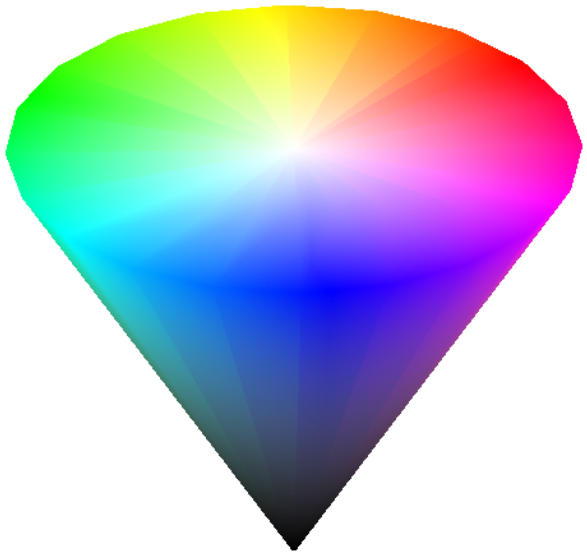
$$S = 1 - \frac{3 \cdot \min(R,G,B)}{R+G+B}, \tag{7}$$

$$H = \begin{cases} \cos^{-1}\left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right]; & Se\ B \leq G \\ 360 - \cos^{-1}\left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right]; & Se\ B > G \end{cases}. \tag{8}$$

2.1.4 Formatos de Arquivos

O principal formato de arquivo *open source* para imagens vetoriais é o SVG (*Scalable Vector Graphics*). Para imagens matriciais os principais formatos são: BMP (*Windows Bitmap*), PNG (*Portable Network Graphics*), GIF (*Graphics Interchange Format*), JPEG (*Joint Photographics Experts Group*). Além destes existem muitos outros formatos para categorias específicas de imagens, como,

Figura 5: Espaço de Cores HSI.



Fonte: Próprio autor

por exemplo, o formato FITS (*Flexible Image Transport System*) para astronomia (29) e o protocolo DICOM (*Digital Imaging and Communications in Medicine*) (30) utilizado em aplicações médicas. A seguir é apresentada resumidamente a descrição de alguns destes formatos.

O formato SVG consiste em um arquivo de texto onde é definido as formas contidas na imagem utilizando a linguagem SVG. A linguagem SVG é uma extensão da linguagem XML (*eXtensible Markup Language*) que tem o objetivo de descrever gráficos bi-dimensionais. Utilizando a linguagem SVG é possível descrever formas vetoriais (linhas retas, curvas, entre outras), texto ou referências para outras imagens. Os objetos gráficos descritos nesta linguagem podem ser agrupados, estilizados, transformados e/ou compostos com outros objetos gráficos descritos na mesma linguagem. As transformações que podem ser enunciadas consistem em máscaras para o canal *alpha* (transparência), filtros, entre outros (31).

O formato *bitmap* consiste em um arquivo binário onde uma imagem matricial é armazenada sem compressão. O arquivo *bitmap* pode ser dividido nas seguintes partes: *i)* o cabeçalho do arquivo onde é armazenado informações gerais como, por exemplo, os dois *bytes* de identificação do formato; *ii)* o tamanho em *bytes*; *iii)* o endereço do primeiro pixel da imagem no ficheiro; *iv)* a palheta de cores onde são definidas as cores utilizadas; *v)* o *array* de pixels onde são armazenados todos os pixels contidos na imagem, da esquerda para direita, de cima para baixo; e *vi)* o perfil de cor utilizado para o gerenciamento de cores (32).

O formato PNG consiste em um arquivo binário no qual uma imagem matricial é armazenada utilizando um algoritmo de compressão *lossless*, ou seja, um algoritmo de compressão onde não ocorre perda de dados. O início do arquivo PNG é composto por: *i)* dois *bytes* de identificação do formato do arquivo, *ii)* seis *bytes* pré-definidos. Em seguida o arquivo contém uma sequência de pedaços

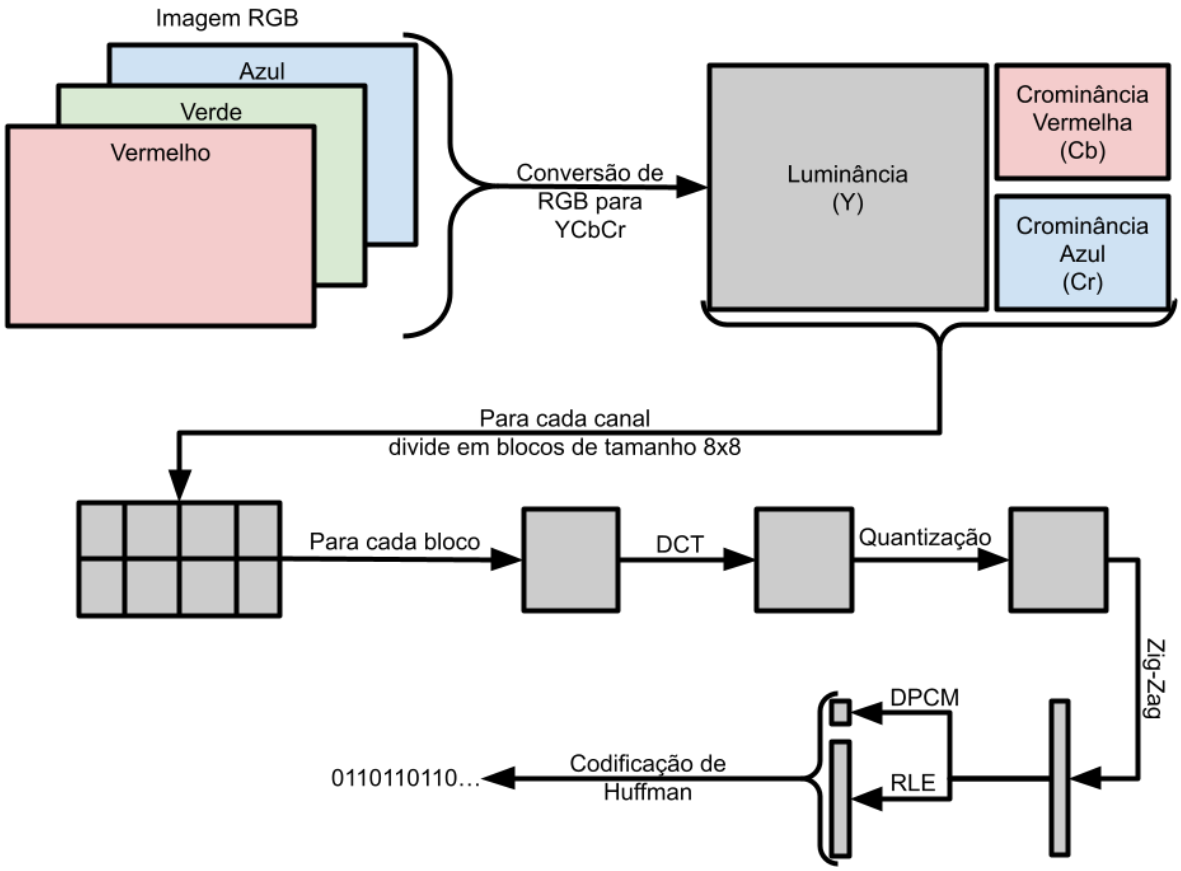
de informações necessárias para representar a imagem, sendo: *i)* paleta de cores, *ii)* valores de cada pixel, *iii)* informação sobre a transparência, *iv)* definição do espaço de cores utilizado, *v)* informações textuais como metadados, *vi)* título, *vii)* autor, *viii)* descrição, entre outros. O método de compressão é derivado do LZ77 (33) usado em programas como zip, gzip, pkzip, para citar alguns (34).

O formato GIF é um formato de arquivo binário utilizado para representar imagens matriciais ou sequências de imagens matriciais que em conjunto definem uma animação. O formato GIF tem uma limitação na quantidade de cores que podem ser representadas na imagem, se limitando a 256 cores pré-definidas. O arquivo GIF é composto por um cabeçalho que contém: *i)* os três *bytes* de identificação do formato do arquivo, *ii)* a descrição lógica da tela, *iii)* a tabela de cores global, *iv)* os valores correspondentes aos pixels das imagens, *v)* o *trailer*, que é o último *byte* definindo o final do ficheiro (32).

O formato JPEG é um formato de arquivo binário utilizado para armazenar imagens matriciais através de um algoritmo de compressão *lossy*, isto é, um método de compressão com perda de dados. Existem algumas variações do formato JPEG como exemplo o JPEG 1, JPEG 2000, JPEG AI, JPEG AIC, entre outras. Neste trabalho será apresentado resumidamente o funcionamento do formato JPEG 1, porém ainda existem algumas variações do formato JPEG 1 como o JPEG 1 *baseline* e o JPEG 1 progressivo. Este trabalho destaca o JPEG 1 *Baseline* especificado na *ITU Recommendation T.81*, pois essa especificação corresponde a primeira implementação proposta para o algoritmo de codificação JPEG que visa somente comprimir e descomprimir imagens. As demais versões, além de atuar na compressão, também garantem que outras propriedades sejam atendidas, tornando o algoritmo mais intrincado.

O algoritmo para armazenar uma imagem RGB em JPEG *Baseline* começa convertendo a imagem RGB para o espaço de cores YCbCr. Em seguida cada canal da imagem é dividido em blocos de tamanho 8×8. Para cada bloco 8×8 é calculado a transformada do cosseno discreto obtendo-se 64 coeficientes. O primeiro coeficiente é denominado DC, e os demais, AC. Cada matriz de coeficientes é dividida elemento a elemento por uma matriz de quantização definida na especificação do formato JPEG e pode ser ajustada de modo a alterar a porcentagem de perda durante a compressão da imagem. É importante ressaltar que a matriz de quantização difere quando se considera a luminância em comparação com a crominância. Isto ocorre por o ser humano ter maior dificuldade em perceber diferenças visuais na crominância. Assim, estes canais da imagem podem sofrer perdas mais significativas durante a compressão. Em seguida as matrizes resultantes destas divisões são percorridas em zig-zag de modo a converter cada matriz para um *array*. Em seguida os coeficientes DC resultantes são comprimidos utilizando a modulação por codificação de pulso diferencial (DPCM), e os coeficientes AC são comprimidos utilizando a codificação *run-length* (RLE). Para finalizar, os valores resultantes da compressão DPCM e RLE são comprimidos novamente utilizando a codificação de *Huffman* e então são armazenados em um arquivo binário conforme a especificação *JPEG File Interchange Format* (JFIF) (35, 36).

Figura 6: Compressão de uma imagem no formato JPEG.

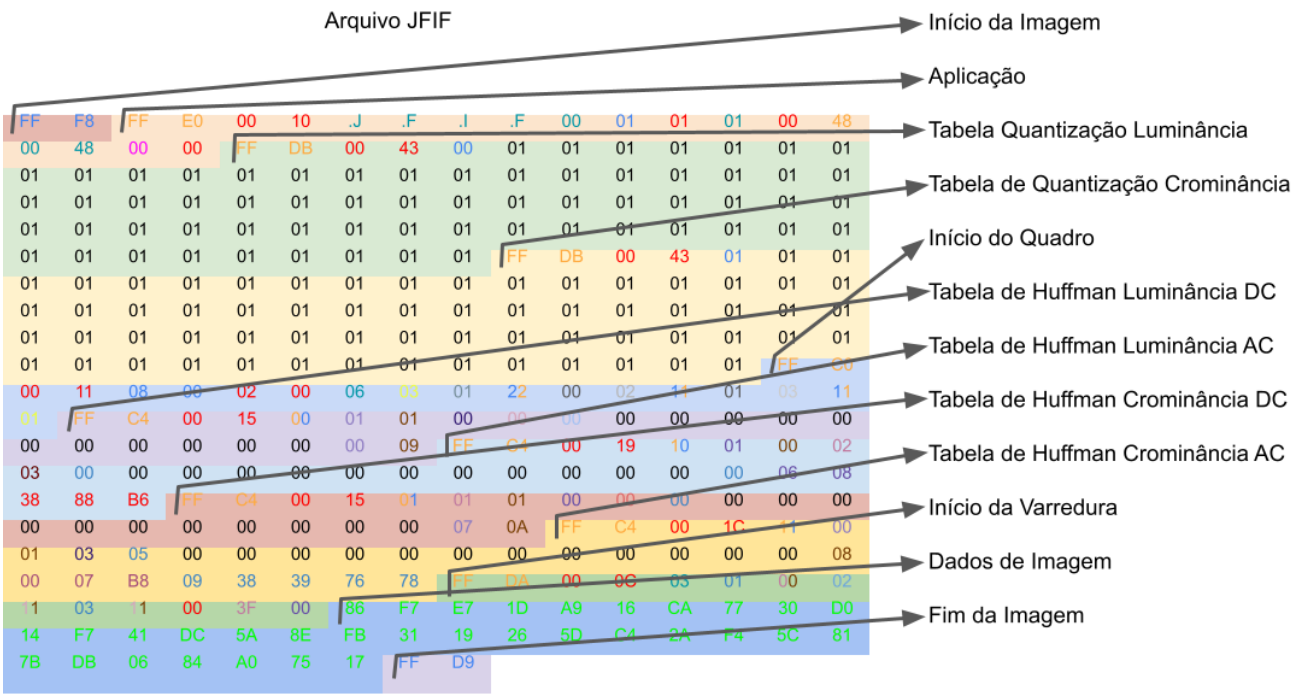


Fonte: Adaptado de (37)

2.2 Criptografia

A criptografia é uma técnica que visa proteger a transmissão ou o armazenamento de uma informação através da codificação desta informação. Atualmente a criptografia é frequentemente utilizada para proteger informações que trafegam ou são armazenadas em servidores conectados à *internet*. Os algoritmos de criptografia podem ser divididos em duas classes, a saber: *i)* criptografia simétrica e *ii)* criptografia assimétrica. Os algoritmos de criptografia simétrica utilizam a mesma chave para codificar e decodificar a informação. Entre estes algoritmos podemos citar: *i)* (*Advanced Encryption Standard*) - AES; *ii)* (*Data Encryption Standard*) - DES; *iii)* (*International Data Encryption Algorithm*) - IDEA; *iv)* (*Drop-in replacement for DES or IDEA*) - *Blowfish*. Os algoritmos de criptografia assimétrica utilizam chaves diferentes para codificar e decodificar a informação, na maioria deles uma chave pública é utilizada para codificar enquanto uma chave privada é utilizada para decodificar. Entre os algoritmos de criptografia assimétrica podemos citar: *i)* (*Rivest Shamir Adleman*) - RSA; *ii)* (*the Digital Signature Standard*) - DSS; *iii)* (*Elliptical Curve Cryptography*) - ECC (38).

Figura 7: Especificação de um arquivo JFIF.



Fonte: Adaptado de (37)

2.3 Codificação de Canal — Detecção e Correção de Erros

Durante a transmissão e armazenamento de informações, em dispositivos digitais, podem ocorrer alguns erros que alteram o conteúdo que está sendo transmitido. Por exemplo, um HD pode sofrer alguma interferência magnética que altere um *bit* de uma informação armazenada no disco. Portanto, para recuperar a informação original e evitar que a mesma seja corrompida é necessário utilizar alguma técnica para identificar e corrigir este *bit*. As técnicas de codificação de canal consistem em um conjunto de algoritmos que visam detectar e/ou corrigir erros em uma sequência de *bits* que está sendo transmitida ou armazenada em algum dispositivo (39).

Inicialmente foi realizada uma revisão da literatura visando determinar quais seriam as técnicas de detecção e correção de erros mais adequadas a serem aplicadas neste trabalho. Nesta revisão foi identificado que o código de *Hamming* e o código de *Reed Solomon* atendem os requisitos necessários para serem utilizados em sistemas de esteganografia digital em imagens, sendo que o primeiro permite a correção de até um *bit* errado, enquanto o segundo permite a correção de uma quantidade maior de *bits* incorretos. É importante ressaltar que embora o código de *Reed Solomon* permita corrigir uma quantidade maior de *bits*, a sua implementação e incorporação ao sistema de esteganografia é mais difícil se comparada ao código de *Hamming*.

Como será demonstrado no capítulo 7, nos experimentos realizados neste trabalho não foram identificados muitos casos onde a mensagem secreta foi recuperada com mais de um *bit* errado, por-

tanto se optou por utilizar o código de *Hamming* para tratar erros que podem ocorrer no processo de recuperação da mensagem armazenada pelo sistema de esteganografia. A seguir são apresentados os dois algoritmos supracitados para detecção e correção de erros (ou seja, o código de *Hamming* e o código de *Reed Solomon*).

2.3.1 Código de *Hamming*

O código de *Hamming* é uma técnica de codificação de canal que permite a detecção e correção de até um *bit* errado através da adição de *n bits* redundantes na informação a ser transmitida ou armazenada. Sendo que o valor de *n* pode ser encontrado a partir da Equação (9), onde *m* representa a quantidade de *bits* necessárias para representar a informação (40).

$$2^n \geq m + n + 1. \tag{9}$$

Para codificar uma informação com o código de *Hamming* é realizado os seguintes passos:

1. A quantidade de *bits* redundantes necessários é calculada através da Equação (9).
   Por exemplo, se a mensagem contém cinco bits então $m = 5$ *e* $n = 4$
2. A sequência de *bits* da informação terá tamanho $m + n$.
3. Os *bits* da informação são deslocados para as posições onde o índice não é uma potência de 2. É importante ressaltar que a contagem dos índices se inicia pelo *bit* menos significativo, mais a direita, e termina no *bit* mais significativo, mais a esquerda. O primeiro *bit* é representado pelo índice 1, enquanto o último pelo índice $n + m$.
   Por exemplo, se a mensagem for 10101 *ela será representada por* _1010_1__
4. As posições nas quais os índices são potências de dois conterão os *bits* redundantes.
   No exemplo anterior são as posições que contém o caracter _
5. A paridade de cada *bit* redundante é calculada e armazenada através da regra a seguir:
   (a) n_1 é o *bit* de paridade para todos os *bits* nos quais os valores binários dos índices incluem o *bit* 1 na posição 1, excluindo-se o índice 1.
   (b) n_2 é o *bit* de paridade para todos os *bits* nos quais os valores binários dos índices incluem o *bit* 1 na posição 2, excluindo-se o índice 2.
   (c) n_i é o *bit* de paridade para todos os *bits* nos quais os valores binários dos índices incluem o *bit* 1 na posição i, excluindo-se o índice i.

   Portanto, na sequência do exemplo ficaria: 010101101

Para identificar e corrigir um erro em uma informação codificada com o código de *Hamming* são realizados os seguintes passos:

1. A quantidade de *bits* redundantes necessários é calculada através da Equação (9).

2. As posições nas quais os índices são potências de dois conterão os *bits* redundantes.
   O exemplo para este passo não será mostrado, pois é idêntico ao mostrado no procedimento anterior.

3. A paridade de cada *bit* redundante é calculada e armazenada através da regra a seguir:

   (a) n_1 é o *bit* de paridade para todos os *bits* nas quais os valores binários dos índices incluem o *bit* 1 na posição 1.

   (b) n_2 é o *bit* de paridade para todos os *bits* nas quais os valores binários dos índices incluem o *bit* 1 na posição 2.

   (c) n_i é o *bit* de paridade para todos os *bits* nas quais os valores binários dos índices incluem o *bit* 1 na posição i.

   Utilizando como entrada a saída do exemplo anterior 010101101*, a sequência de bits resultante do procedimento será:* 010100100

4. Os *bits* de redundância representam um número binário x que será convertido para o formato decimal.

   (a) Se $x = 0$ não existe erro de um *bit*.

   (b) Se $x > 0$ então, o *bit* x apresenta o valor errado. Caso exista apenas um *bit* errado, invertendo o *bit* na posição x a informação original é recuperada.

   Considerando a saída do item anterior 010100100*, extraindo apenas os bits de redundância encontramos a sequência de bits* 0000 *que em decimal representa o valor* 0*, ou seja, não existe erro de bit.*

Se mais de um *bit* apresentar erro o procedimento de correção não funcionará. Para verificar se a informação apresentava apenas um erro, após a primeira aplicação do código de *Hamming* basta executar novamente o procedimento de identificação e correção de erro em uma informação e verificar se $x = 0$ apresenta o valor verdadeiro. Caso esta condição seja satisfeita, o código de *Hamming* conseguiu identificar e corrigir o erro, caso contrário não é possível corrigir o erro, pois a informação apresenta mais de um *bit* errado (40).

2.3.2 Código de *Reed Solomon*

O código de *Reed Solomon* é uma técnica de codificação de canal baseada em corpos de *Galois* que permite a detecção e correção de erros em uma sequência de *bits* através da adição de alguns *bits* de paridade. A quantidade máxima de erros que poderá ser detectada e corrigida pode ser determinada

no momento da aplicação da técnica. Contudo, quanto maior for o número de *bits* a ser corrigido, maior será o número de *bits* de paridade a ser adicionado à sequência de *bits* original. Utilizando esta técnica, ao adicionar t *bits* de paridade a uma sequência de *bits* é possível detectar e corrigir $\frac{t}{2}$ erros em posições desconhecidas ou t erros em posições conhecidas (41).

O código de *Reed Solomon* divide a sequência de símbolos a ser codificada em blocos de 255 símbolos. Assim, ao se escolher utilizar *bits* como unidade, a informação é dividida em blocos de 255 *bits*. No entanto, ao se escolher utilizar *bytes* a informação é dividida em blocos de 255 *bytes*, e assim por diante. Cada bloco conterá n símbolos para mensagem e m símbolos de paridade, sendo que a configuração escolhida pode ser descrita no seguinte formato: $R-S(n,m)$ (41).

A recuperação da informação utilizando o código de *Reed Solomon* pode levar aos seguintes resultados:

- Informação recebida sem erros.
- Informação recebida com erros, mas é possível recuperar a informação original.
- Informação recebida com erros, e não é possível recuperar a informação original.
- Informação recebida com mais símbolos errados do que o valor máximo suportado pela implementação, neste caso o erro pode não ser percebido.

O código de *Reed Solomon* é amplamente utilizado em diversas aplicações, como, por exemplo, *QR Codes*, CDs, DVDs, *storages* (42), comunicações com sondas espaciais no espaço profundo como na missão *Voyager2* lançada pela NASA em 1977 (43).

2.4 Medidas de Semelhanças entre Imagens

Alguns métodos podem ser empregados para comparar duas imagens, aferindo o quão distantes estas se encontram. Estes métodos podem ser baseados em aspectos matemáticos como as métricas MSE (*Mean Squared Error*), PSNR (*Peak Signal to Noise Ratio*) e SSIM (*Structural Similarity*), ou com base em simulações da percepção visual humana, como as métricas perceptuais. A seguir será apresentada sucintamente cada métrica citada.

2.4.1 Erro Quadrático Médio - MSE

O MSE representa o erro quadrático médio entre a imagem original e a imagem modificada. Maiores valores para o MSE representam uma diferença maior entre as imagens. O valor de MSE pode ser obtido através da Equação (10).

$$\text{MSE} = \frac{1}{MN}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\left(f(x,y)-\tilde{f}(x,y)\right)^2, \tag{10}$$

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 (44).

2.4.2 Relação Sinal-Ruído de Pico - PSNR

O PSNR representa uma medida de erro de pico. Maior valor de PSNR constitui menor diferença entre as imagens. O valor de PSNR é calculado através da Equação (11)

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_i^2}{\text{MSE}} \right), \tag{11}$$

onde MSE representa o valor calculado pela Equação (10) e MAX_i^2 representa o maior valor para um pixel presente na imagem, geralmente esse valor é 255 (44).

2.4.3 Medida do Índice de Similaridade Estrutural - SSIM

O SSIM é um método utilizado para mensurar a qualidade percebida em imagens e vídeos digitais. Maior valor de SSIM representa menor diferença entre as imagens. O valor de SSIM é calculado através da Equação (12):

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{12}$$

onde μ_x e μ_y representam os valores médios de x e 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 $k_2 = 0,03$ (44).

2.4.4 Métricas Perceptuais

As métricas perceptuais visam mensurar o erro percebido por um ser humano ao realizar a comparação de duas imagens. Dado que o sistema visual humano é complexo e difícil de ser modelado, não é simples definir tais categorias de métricas. Algumas métricas perceptuais utilizam redes neurais convolucionais visando mensurar a capacidade de um ser humano diferenciar duas imagens, sendo que no treinamento destas redes são utilizados *datasets* gerados a partir da comparação de diversas imagens analisadas por pessoas distintas. Após estas redes serem treinadas elas são usadas para prever a capacidade de um ser humano perceber diferenças entre duas imagens. Estas métricas também podem ser utilizadas para prever qual imagem uma pessoa escolheria caso precisasse selecionar, entre duas imagens modificadas, qual se assemelha mais à imagem original (45, 46). Para este trabalho foi escolhida a métrica LPIPS — *Similaridade Perceptual*, por apresentar embasamento na literatura e pela facilidade de aplicar esta métrica às imagens pós-processadas pelo sistema de esteganografia utilizando-se apenas uma biblioteca para a linguagem *Python* (46).

3 Redes Neurais Artificiais

Neste capítulo são discutidos alguns conceitos e definições relacionados ao tema de redes neurais artificias que são importantes para este trabalho. Assim será apresentado: *i)* uma breve introdução ao tema, *ii)* a definição de um neurônio artificial, *iii)* o conceito de rede neural artificial, *iv)* funções de ativação, *v)* arquiteturas de redes, *vi)* o conceito de uma rede neural convolucional, *vii) autoencoders*, e *viii)* redes neurais adversárias generativas.

3.1 Introdução

Para definir o que é Inteligência Artificial (IA) é necessário anteriormente definir inteligência. Na literatura existem diversas definições de inteligência. Em (47) são apresentadas algumas delas, as quais são classificadas como: *i)* coletivas, *ii)* da psicologia, *iii)* provenientes da pesquisa sobre inteligência artificial. A partir de (47), foram selecionadas duas definições relevantes relacionadas ao objetivo deste trabalho, as quais são apresentadas a seguir. A partir da psicologia pode-se definir inteligência como: "aquela faceta da mente subjacente à nossa capacidade de pensar, resolver novos problemas, raciocinar e ter conhecimento do mundo" (48). Refletindo o ponto de vista da pesquisa em inteligência artificial, a ideia de inteligência pode ser expressa como: "A capacidade de um sistema agir adequadamente em um ambiente incerto, onde a ação apropriada é aquela que aumenta a probabilidade de sucesso, e o sucesso é a realização de submetas comportamentais que suportam o objetivo final do sistema" (49).

Caminhando para definição de inteligência artificial, nos deparamos novamente com uma extensa variedade de expressões presentes na literatura. A primeira delas foi apresentada em 1950 por Alan Turing no trabalho "*Computing Machinery and Intelligence*", a qual define inteligência artificial como: "A ciência e engenharia de fazer máquinas inteligentes, especialmente programas de computador inteligentes" (50). Outra definição mais recente e próxima do contexto deste trabalho foi proposta em 2004 por John McCarthy, com o conceito sendo definido como: "É a ciência e engenharia de elaborar máquinas inteligentes, especialmente programas de computador inteligentes. Está relacionado à tarefa semelhante de usar computadores para entender a inteligência humana, mas a IA não precisa se limitar a métodos biologicamente observáveis" (51).

Atualmente é possível encontrar aplicações da inteligência artificial em diversos setores, como, por exemplo: comunicações, gerenciamento de tempo, saúde, segurança, educação, jogos, entretenimento, *marketing*, vendas, experimentos científicos, desenvolvimento de medicamentos, previsões de eventos climáticos, agricultura, engenharia, arquitetura, transporte, finanças, música, aviação e ampliação da cognição humana (52).

Os sistemas de IA podem ser classificados em: *i)* IA fraca ou *ii)* IA forte. Esta classificação foi proposta pelo filósofo John Searle em 1980. De acordo com Searle, a inteligência artificial fraca constitui uma ferramenta muito poderosa para realização de testes de hipóteses e simulações rigorosas

e precisas. Já na inteligência artificial forte o computador é programado para se comportar como uma inteligência de propósito geral com estados cognitivos (53). A IA fraca é geralmente implementada para aprender a lidar com tarefas específicas (54), enquanto a IA forte deve conseguir aprender a lidar com qualquer tarefa (55).

A inteligência artificial forte pode ser entendida como uma máquina que possui uma mente composta por aplicações sendo executadas em algum equipamento físico e não em condições advindas da biologia (53). Atualmente não existe na literatura evidências de que algum sistema de inteligência artificial forte tenha sido implementado (55).

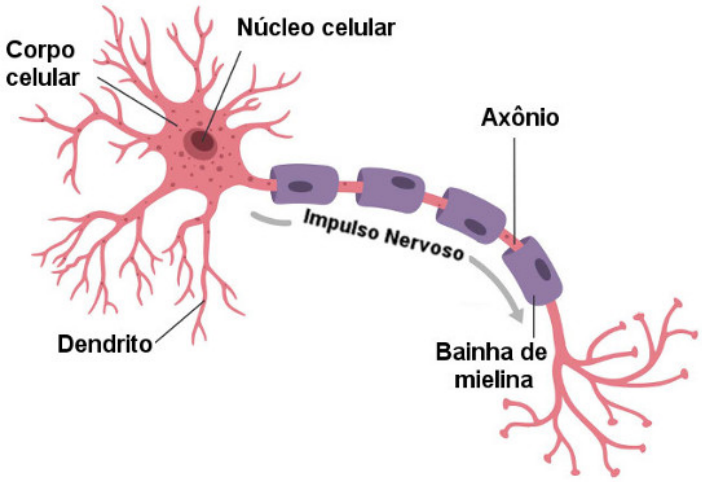
Os algoritmos de aprendizado de máquina utilizados para implementar sistemas de inteligência artificial fraca podem ser classificados em quatro abordagens: *i)* simbólica, *ii)* probabilística, *iii)* conexionista, e *iv)* evolutiva. A abordagem simbólica é baseada na inferência de um conjunto de regras a partir de um conjunto de dados durante o treinamento. Após a conclusão do treinamento o sistema utiliza estas regras para tomada de decisão. As regras que compõem o sistema são comumente organizadas na estrutura de árvore de decisão (56). A abordagem probabilística utiliza algoritmos baseados em teoremas probabilísticos. Como exemplo podemos citar o classificador *Naïve Bayes* baseado no teorema de Bayes, abordagem utilizada em problemas de classificação (57). A abordagem conexionista é baseada na simulação das redes neurais biológicas através da utilização de modelos matemáticos de neurônios. É a principal abordagem utilizada em sistemas de inteligência artificial atualmente. Esta abordagem foi criada em 1943 por McCulloch e Pitts com a proposta do primeiro modelo matemático de neurônio (56). A abordagem evolutiva se baseia na utilização de algoritmos genéticos para auxiliar no treinamento de redes neurais. Os algoritmos genéticos são uma classe de algoritmos probabilísticos inspirados na teoria Darwiniana da evolução das espécies. Estes algoritmos fornecem um mecanismo de busca paralela e adaptativa (58).

3.2 Redes Neurais Biológicas

Redes Neurais biológicas, como a rede neural humana, funcionam a partir de células denominadas neurônios. O cérebro humano possui cerca de 10^{11} neurônios e, embora existam diversos tipos deles, a maioria compartilha os mesmos recursos que são: *i)* as ramificações de dendritos, onde ocorre a entrada de sinais para o neurônio, *ii)* o corpo celular, onde é realizado o processamento, e *iii)* o axônio, que é onde ocorre a saída deste processamento através de uma sinapse. A transferência de informação entre dois ou mais neurônios ocorre em um processo denominado sinapse. Cada neurônio normalmente tem conexões com milhares de outros neurônios. Desse modo, o total de sinapses no cérebro humano excede 10^{14}. Embora o neurônio realize a tarefa de processar informações lentamente, a abundância de neurônios no cérebro leva a um grande paralelismo, que torna o processamento da rede neural eficiente. Após processar um sinal recebido nos dendritos, o neurônio pode ou não liberar outro sinal através do axônio, sendo que caso ocorra a liberação, a sinapse sucedida pode ser inibitória ou excitatória. Sinapses inibitórias aumentam a probabilidade dos neurônios adjacentes não transmitirem

outro sinal, enquanto sinapses excitatórias efetuam o inverso (59).

Figura 8: Neurônio biológico.



Fonte: Retirada de (60)

Inspirados nas redes neurais biológicas, surgiu a classe de algoritmos denominados redes neurais artificiais, que fornece um novo paradigma para processamento de dados em computadores sem precisar da especificação de regras bem definidas sobre como os dados serão tratados durante seu processamento (59).

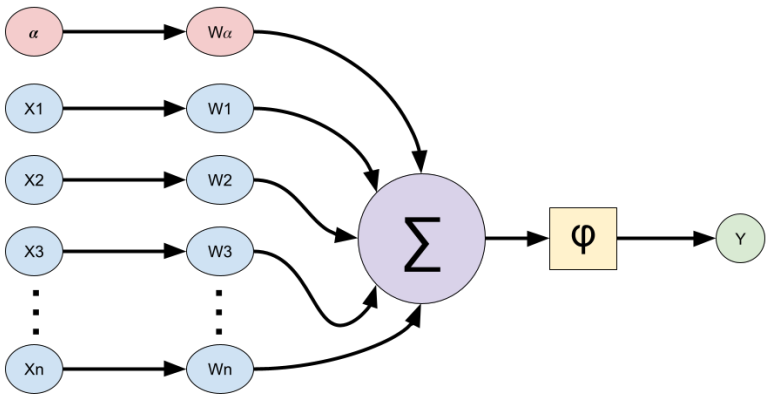
3.3 Modelos de Neurônios

Redes neurais artificiais consistem em um conjunto de neurônios artificiais como, por exemplo, o neurônio de McCulloch. O neurônio de McCulloch, representado na Figura 9, tem a função de receber um conjunto de variáveis $x_1, x_2, x_3, ..., x_n$, atribuir um peso $w_1, w_2, w_3, ..., w_n$ a cada variável, processá-las por uma função de ativação e retornar uma variável y como saída. A função de ativação utilizada pelo neurônio de McCulloch é definida na Equação 13 (59, 61).

$$y = \sum_{i=1}^{n} w_i x_i + \alpha, \tag{13}$$

onde y representa a saída, x_i a i-ésima entrada, e w_i o i-ésimo peso sináptico do respectivo neurônio. Em uma rede neural, como a representada na Figura 10, cada neurônio tem seus próprios pesos sinápticos. O parâmetro α representa o viés de um neurônio específico. Numa analogia com o neurônio biológico, o valor para o viés corresponderia ao limiar do respectivo neurônio liberar um sinal após a conclusão de seu processamento (61). O algoritmo de rede neural artificial simula as sinapses entre um conjunto de neurônios artificiais. A definição dos valores dos pesos sinápticos é realizada em um processo denominado treinamento. Para o treinamento de uma rede neural podem ser utilizados diversos algoritmos, por exemplo: o método do gradiente descendente, o método de Newton, o algoritmo

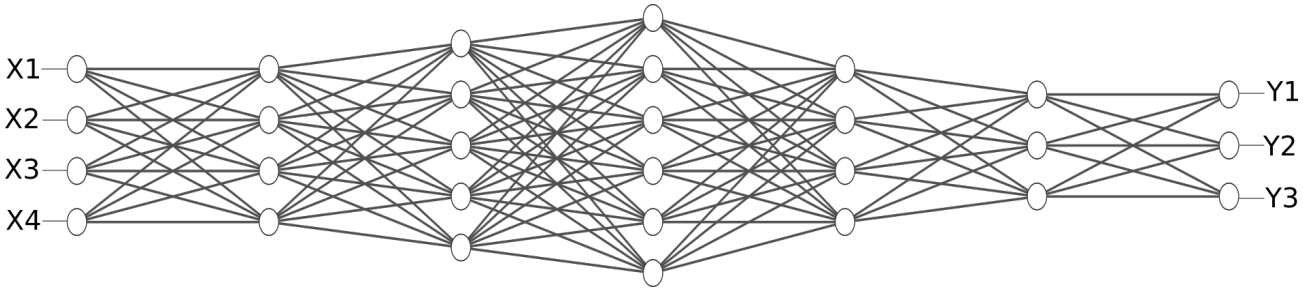
Figura 9: Neurônio de McCulloch.



Fonte: Adaptado de (62)

de Levenberg-Marquardt, entre outros (59).

Figura 10: Rede Neural Artificial.



Fonte: Gerado através da ferramenta *NN-SVG*[1]

3.4 Funções de Ativação

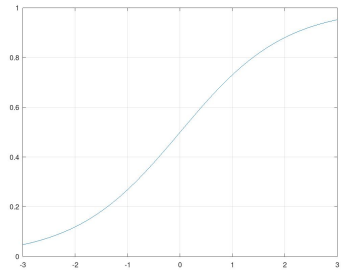
As funções de ativação são responsáveis por processar a combinação linear das entradas e dos pesos sinápticos para cada neurônio que compõe a rede neural. Após o processamento, a função de ativação retorna o sinal de saída do neurônio (63, 64). Durante o treinamento da rede neural artificial utilizando o algoritmo clássico *backpropagation* (em português, retropropagação) é necessário calcular a derivada do erro em relação aos pesos de cada camada, indo da última para primeira camada e efetuar ajustes nos pesos sinápticos para minimizar o erro. Portanto, a função de ativação deve ser derivável (65, 64), exceto talvez em um conjunto de pontos de medida nula.

Existem diversas funções de ativação presentes na literatura. Algumas das funções de ativação mais comuns são: *i)* Sigmoide (Equação (14)), *ii)* Tangente Hiperbólica (Equação (15)), *iii)* Linear (Equação (16)), *iv) Softmax* (Equação (17)), *v) Rectified Linear Unit* (ReLU) (Equação (18)), *vi) Scaled Exponential Linear Units* (SeLU) (Equação (19)) (64). Na Figura 11 é apresentado o gráfico

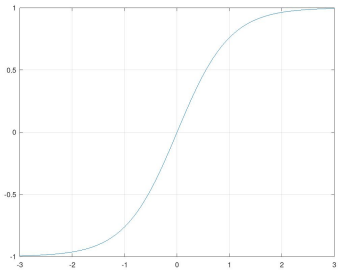
[1] https://alexlenail.me/NN-SVG/

de cada função de ativação citada. Neste trabalho optou-se por utilizar a função de ativação SeLU pelos resultados satisfatórios apresentados nos experimentos realizados.

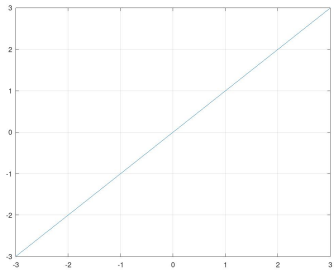
Figura 11: Exemplos de funções de ativação.



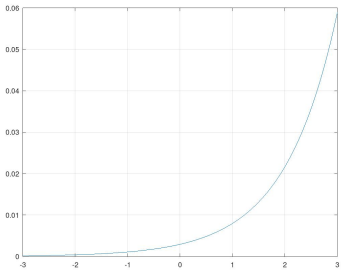
(a) Sigmoide.



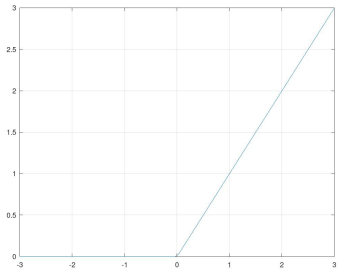
(b) Tangente Hiperbólica.



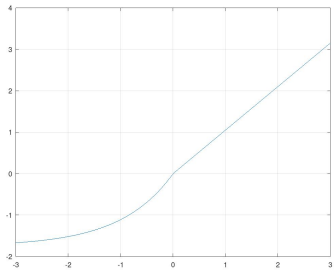
(c) Linear.



(d) *Softmax*.



(e) ReLU.



(f) SeLU.

Fonte: Adaptado de (66)

$$f(x) = \frac{1}{1+e^{-\beta x}}, \tag{14}$$

$$f(x) = tanh(\beta x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}, \tag{15}$$

$$f(x) = x, \tag{16}$$

$$f(x) = \frac{e^x}{\sum e^x}, \tag{17}$$

$$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases}, \tag{18}$$

$$\begin{gathered} f(x) = \lambda \begin{cases} \alpha(e^x - 1), & x \leq 0 \\ x, & x > 0 \end{cases} \\ \lambda = 1.0507 \\ \alpha = 1.6733 \end{gathered} \tag{19}$$

3.4.1 Sigmoide

A função sigmoide (Equação (14)) é a mais comumente utilizada em problemas de aprendizado de máquina. É uma função de ativação não linear que retorna valores no intervalo entre 0 e 1. É uma função contínua e diferenciável. Seu gráfico se assemelha suavemente ao formato da letra S. A função sigmoide não é simétrica em relação à origem (66).

3.4.2 Tangente Hiperbólica

A função tangente hiperbólica (Equação (15)) é similar a função sigmoide, porém é simétrica em relação à origem. Além disto, os valores retornados encontram-se no intervalo entre -1 e 1. É uma função contínua e diferenciável (66).

3.4.3 Linear

A função linear (Equação (16)) é diretamente proporcional à entrada. Não existem muitos benefícios para se utilizar está função de ativação em redes neurais, dado que o erro da rede neural não será reduzido ao obter-se o mesmo valor para o gradiente em todas as iterações. A utilização desta função de ativação também reduz a capacidade da rede neural identificar padrões mais complexos nos conjuntos de treinamento, porque diversas camadas lineares podem ser equivalentemente descritas como uma única camada linear. Esta função é idealmente utilizada em tarefas simples que exijam interpretabilidade (66).

3.4.4 Softmax

A função softmax (Equação (17)) consiste em uma combinação de múltiplas funções sigmoide. Esta função de ativação é comumente utilizada em problemas de classificação multiclasse. É importante ressaltar que quando uma rede neural artificial é construída para resolver problemas de classificação multiclasse, o número de neurônios da última camada da rede neural deve ser igual à quantidade de classes (66).

3.4.5 ReLU

A função ReLU (Equação (18)) consiste em uma função de ativação não linear, utilizada amplamente em problemas de aprendizado de máquina. Uma das vantagens de utilizar a função ReLU é a implicação de que nem todos os neurônios que compõem a rede neural estarão ativados simultaneamente. (66)

3.4.6 SeLU

A função SeLU é semelhante à função ReLU, com a diferença de introduzir o conceito de auto-normalização. Ao contrário da função ReLU, a função SeLU pode apresentar saídas menores

que zero. (67)

3.5 Formas de Aprendizado

Os algoritmos de aprendizado de máquina podem ser classificados em: *i)* supervisionados e *ii)* não-supervisionados, *iii)* por reforço. Os algoritmos supervisionados aprendem a partir de conjuntos de dados previamente rotulados. Estes algoritmos são utilizados para a execução de tarefas específicas como, por exemplo, em problemas de classificação ou regressão (68). Já os algoritmos não-supervisionados aprendem a partir de conjuntos de dados não rotulados, sendo utilizados em problemas de detecção de anomalias, sistemas de recomendação, clusterização ou agrupamento (69). Os algoritmos de aprendizado por reforço são utilizados em problemas que envolvem tomada de decisão. Estes algoritmos aprendem com base nos resultados de suas decisões através de um processo de tentativa e erro. Durante o treinamento, caso o algoritmo decida executar uma ação apropriada ele recebe uma recompensa, caso contrário recebe uma penalidade. Com isto espera-se que o algoritmo aprenda com seus erros e tome melhores decisões a medida que o treinamento avança (70).

3.6 Arquiteturas de Redes

As Redes Neurais Artificiais (RNA) podem ser implementadas utilizando-se diversas arquiteturas, com cada arquitetura apresentando suas especificidades, vantagens e desvantagens. As principais arquiteturas de RNA são redes: *i) Multilayer Perceptrons*, *ii)* Convolucionais, *iii)* Recorrentes, *iv)* Autoencoders, *v)* Adversárias Generativas (71).

As redes neurais *Multilayer Perceptron* são redes neurais compostas por mais de uma camada, sendo que cada camada é composta por um ou mais perceptrons. Estas redes apresentam uma camada de entrada onde são recebidos os sinais, n camadas intermediárias que realizam o processamento, e uma camada de saída, a qual retorna a decisão ou a previsão sobre a entrada. As redes *Multilayer Perceptron* são utilizadas em problemas de classificação, regressão, entre outros (72).

As redes neurais convolucionais ou redes neurais convolucionais profundas são amplamente utilizadas em problemas de aprendizado de máquina que envolvam processamento de imagens, como, por exemplo, classificação de imagens, clusterização de fotos, detecção de objetos em fotografias. Estas redes também são utilizadas em algoritmos de OCR (*optical character recognition*) para extração de texto em imagens, e em reconhecimento de áudio a partir da análise do espectrograma do áudio. O espectrograma consiste em uma forma de representar um áudio como uma imagem (14).

As redes neurais recorrentes são projetadas para considerar a ordem que os dados são apresentados para rede durante o seu processamento. Estas redes são comumente utilizadas no processamento de áudio, séries temporais e linguagem natural. Em redes neurais recorrentes a saída do instante n é reutilizada como parte da entrada no instante $n+1$. Portanto, nestas redes a saída da última etapa irá influenciar no processamento da próxima etapa. Deste modo, a rede consegue entender uma sequência considerando a sua ordenação (73).

Já os autoencoders são redes compostas por duas metades simétricas. A primeira metade, denominada *encoder*, é responsável por codificar o conjunto de dados da entrada em uma dimensão menor. Já a segunda metade, denominada *decoder*, é responsável por representar, em sua dimensão original, o conjunto de dados produzido na saída da primeira metade da RN. Os autoencoders são utilizados em tarefas como redução de ruído em imagens e melhoria da qualidade de imagens (74).

As arquiteturas baseadas em redes neurais adversárias generativas são compostas por duas redes: geradora e discriminadora, que competem durante o treinamento, por isto sendo denominadas de adversárias. A RN discriminadora tem o objetivo de classificar se uma determinada entrada é original ou sintetizada. Esta rede pode receber como entrada dados provenientes do conjunto original ou provenientes da saída da RN geradora. A rede geradora tem o objetivo de produzir dados sintetizados que se pareçam o máximo possível com os dados originais. O treinamento destas duas redes ocorrem simultaneamente e, à medida que o treinamento avança, a rede neural geradora aprende a "enganar" a rede discriminadora, retornando dados cada vez mais semelhantes aos dados presentes no conjunto original (75, 76).

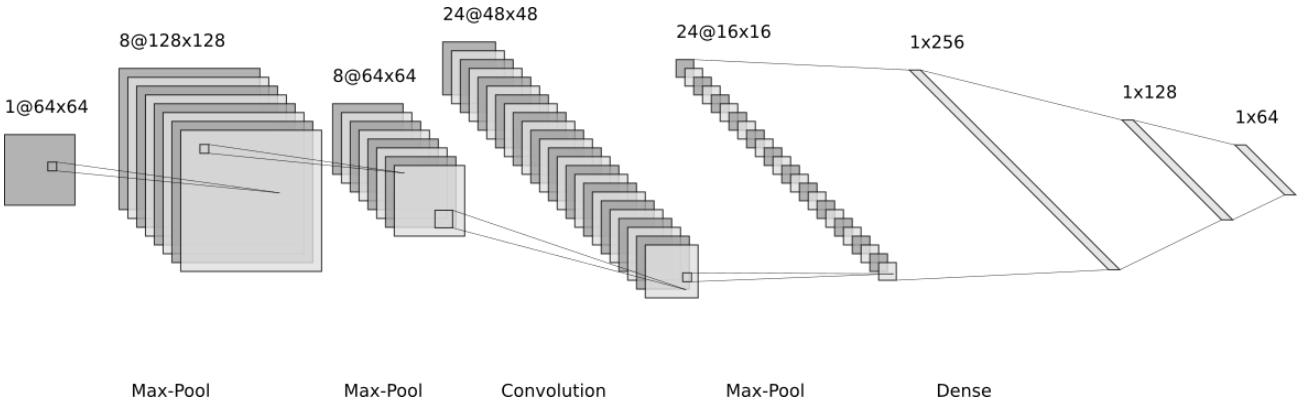
3.7 Redes Neurais Convolucionais

Redes neurais convolucionais (CNN, do inglês, *Convolutional Neural Network*), como a representada na Figura 12, consistem em uma classe de RNA voltada para a detecção de padrões em dados que apresentem dependências espaciais, como imagens. Enquanto a RNA consiste em uma série de camadas de neurônios que propagam suas sinapses até que a saída seja liberada na última camada da rede neural, a CNN realiza seu processamento através de convoluções. Uma convolução é uma operação matemática que permite combinar dois sinais para engendrar um terceiro sinal. Mais especificamente, para CNNs, podemos definir convolução como o processo de adicionar cada elemento da imagem aos seus vizinhos locais, ponderados pelo filtro (*kernel*). Assim, em cada convolução uma região da imagem é processada. Uma característica relevante da CNN consiste na realização do seu processamento utilizando menos parâmetros em relação à redes *multilayer perceptron*. Além disso, a CNN tem se mostrado mais eficiente do que a rede *multilayer perceptron* em uma série de tarefas como: *i)* classificação de imagens e sons, *ii)* geração de imagens, *iii)* geração de sons, *iv)* detecção de objetos, *v)* tradução imagem a imagem, dentre outras. A CNN pode ou não ser utilizada em conjunto com uma rede *multilayer perceptron* (77).

3.8 Autoencoder

Os autoencoders, como o representado na Figura 13, correspondem a uma classe de RNA que tem o objetivo de receber como entrada um conjunto de valores x e retornar na saída valores idealmente idênticos (embora na prática sejam próximos) aos dados de entrada recebidos. A informação retornada pelo autoencoder pode ser denominada por $\bar{x}$. A topologia de um autoencoder é composta por duas redes neurais artificiais. A primeira tem a função de receber a entrada x e reduzir sua dimensão,

Figura 12: Rede Neural Convolucional.



Fonte: Gerado através da ferramenta *NN-SVG*[2]

mantendo os aspectos mais importantes para sua reprodução. Tal redução poderia ser entendida como uma espécie de "compressão neural". A segunda rede neural tem o objetivo de receber esta informação com a dimensionalidade reduzida e retornar uma saída que seja o mais próxima possível da entrada da primeira rede, com a mesma dimensão (74).

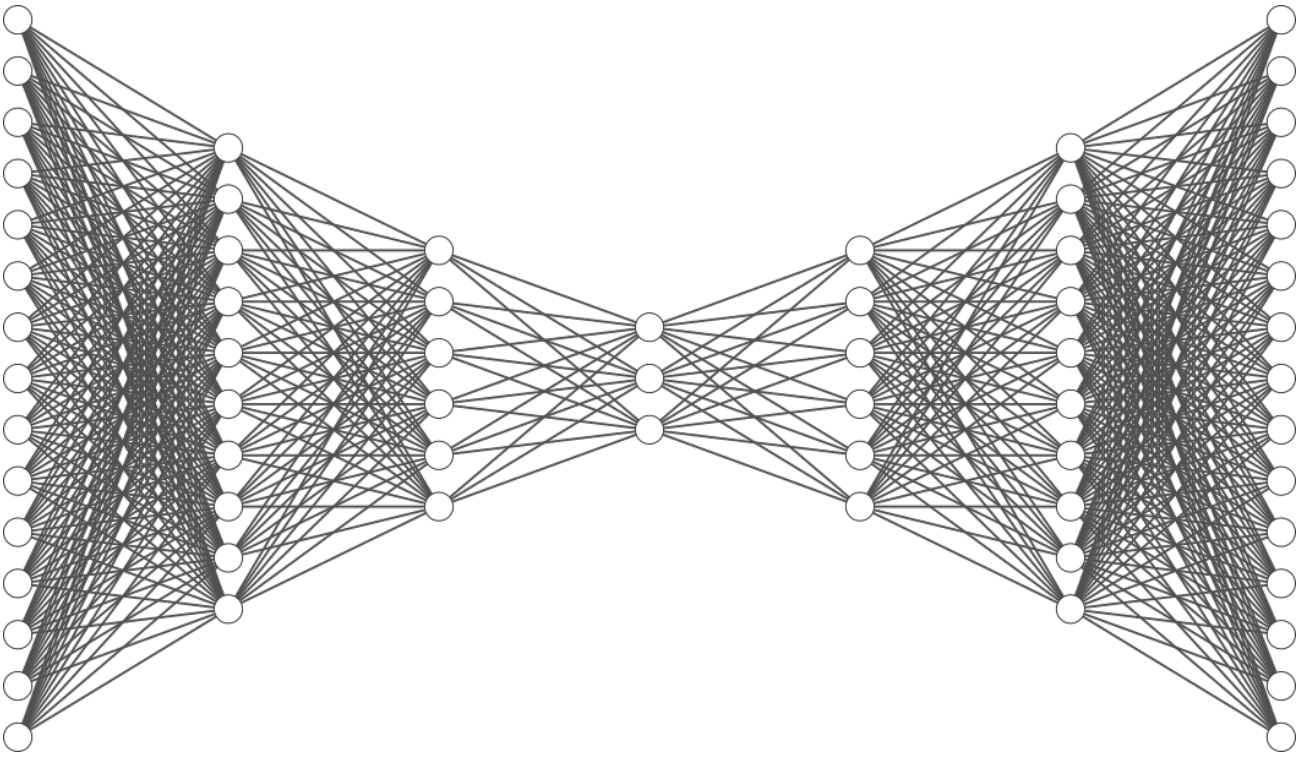
Por exemplo, em um autoencoder composto por duas redes neurais convolucionais, a primeira rede neural deve ser responsável por reduzir a dimensão de uma imagem de 256×256 pixels para uma imagem de 32×32 pixels, enquanto que a segunda rede neural será responsável por receber a informação redimensionada e retornar uma figura próxima à original com dimensão 256×256 pixels. É importante ressaltar que um autoencoder não realiza o redimensionamento da imagem, e sim sua compressão, portanto a imagem em sua dimensão comprimida não será necessariamente uma representação em menor escala da original (74).

3.9 Rede Adversária Generativa

Redes neurais adversárias generativas (GANs), como a representada na Figura 14, consistem em um método de aprendizado de máquina não supervisionado onde duas redes neurais são colocadas para competir uma com a outra. A primeira rede neural é denominada geradora enquanto que a segunda rede neural é denominada discriminadora. O treinamento ocorre a partir de um conjunto de dados reais que pode ser formado por um conjunto de imagens, áudios, entre outros. A rede neural geradora recebe um ruído como entrada e tem o objetivo de gerar como saída um dado sintético semelhante aos dados que compõem o conjunto de dados reais. A rede neural discriminadora recebe como entrada um valor extraído do *dataset* real ou uma saída produzida pela rede neural geradora. Esta rede tem o objetivo de classificar esta informação em real ou sintético. O treinamento das duas redes neurais é realizado simultaneamente, sendo que durante o treinamento a rede neural geradora e a rede neural discriminadora são adversárias. Enquanto a RN geradora aprende a partir do ruído, e ela

[3]https://alexlenail.me/NN-SVG/

Figura 13: Autoencoder.



Fonte: Gerado através da ferramenta *NN-SVG*[3]

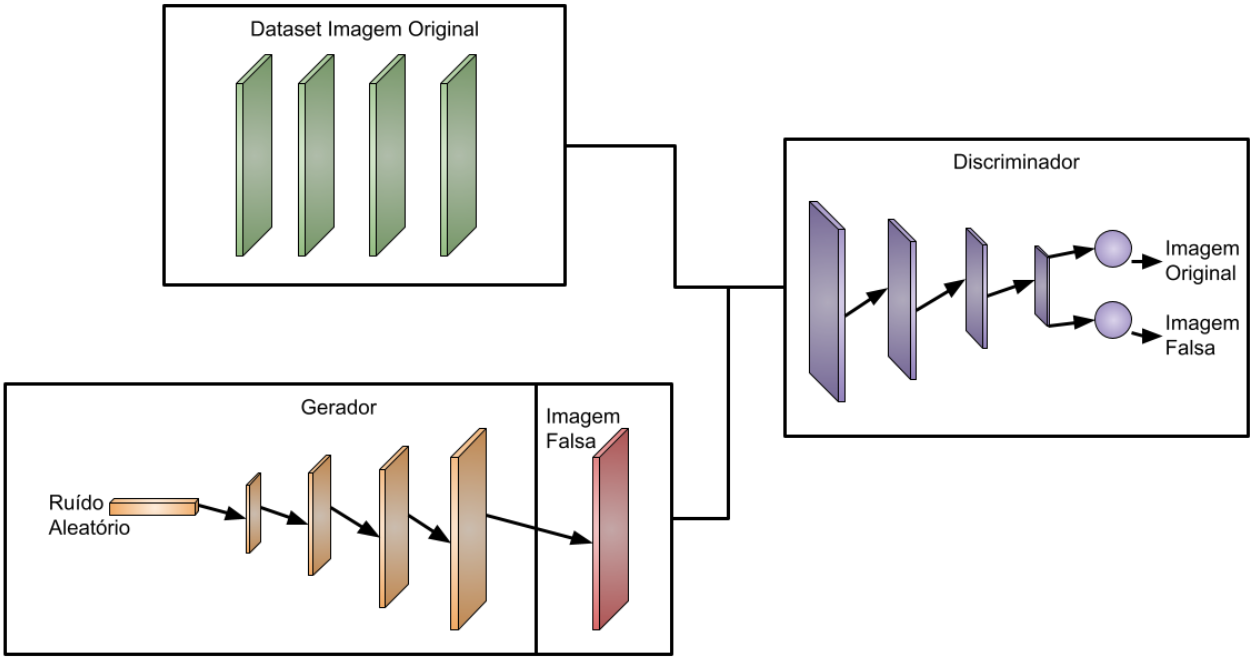
deve gerar uma representação que engane a rede discriminadora em sua tarefa, a RN discriminadora aprende as características que diferem um dado sintético de um dado real. Ao final do treinamento é esperado que a rede neural geradora consiga gerar dados sintéticos tão próximos dos dados reais tal que seja muito difícil para um ser humano ou uma rede neural detectar que se trata de uma informação sintetizada (76).

3.10 Comentários Finais

Neste capítulo, alguns conceitos importantes concernentes à arquitetura e ao treinamento de redes neurais foram concisamente descritos. Como visto, a proposta deste trabalho conjuga redes neurais com esteganografia. Antes que possamos explicitar com maiores detalhes a proposta, é necessário abordar os conceitos de esteganografia e esteganálise com maior cuidado. Tal é o objetivo do próximo capítulo.

[3]https://alexlenail.me/NN-SVG/

Figura 14: Rede Neural Adversária Generativa.



Fonte: Adaptado de (76)

4 Esteganografia e Esteganálise

Esteganografia consiste em uma técnica que permite armazenar de forma oculta uma informação nas partes redundantes de outra informação. Esta técnica foi empregada diversas vezes ao longo da história da humanidade, em ocasiões onde se fez necessário transmitir uma mensagem por um canal público ou perigoso e era extremamente importante que a mensagem transmitida não fosse interceptada, evitando que pessoas não autorizadas tivessem acesso a tal informação (78).

Um exemplo prático de uma informação transmitida com esteganografia consiste na tinta invisível, composta por um pigmento que só pode ser visualizado pelo olho humano quando o papel está na frente de uma fonte de luz ultra-violeta. Para armazenar uma mensagem utilizando esteganografia através desta técnica pode-se escrever, em uma folha de papel branco, a mensagem a ser escondida utilizando a tintura invisível, e com uma tinta normal escrever outra informação que poderá ser revelada caso alguém não autorizado tenha acesso ao papel. Idealmente, caso o papel seja interceptado, quem estiver de posse do mesmo, visualizará a mensagem escrita com a tinta tradicional. Assim, a mensagem secreta permanecerá protegida, caso o papel chegue ao seu destinatário, o mesmo saberá que necessita utilizar uma luz ultra-violeta para ler a informação original.

A esteganálise consiste no conjunto de técnicas desenvolvidas para identificar a existência ou inexistência de uma informação escondida na informação pública em análise. Esta é uma área de estudo bem abrangente quando se trata de identificar emprego da esteganografia em meios digitais. Existem na literatura diversas abordagens para esta categoria de análise, sendo que para cada conjunto de métodos de esteganografia existe um algoritmo de esteganálise ideal. Vale evidenciar que dada a grande diversidade de métodos de esteganografia presentes na literatura, engendrar um único algoritmo de esteganálise que funcione bem para grande maioria das técnicas é uma tarefa difícil, portanto os algoritmos de esteganálise que tentam suportar diversos métodos de esteganografia apresentam usualmente resultados inferiores aos algoritmos que visam atacar um conjunto limitado de técnicas (79).

4.1 Esteganografia Digital

Com o crescente desenvolvimento dos computadores surgiu a esteganografia digital. Tais técnicas permitem ocultar a existência de um dado digital, por exemplo, um arquivo de computador. Para isto são utilizadas técnicas que permitem armazenar o arquivo a ser escondido nas partes redundantes de outro arquivo digital. As categorias de arquivos que são mais utilizados para esteganografia digital são os arquivos que possuem maior quantidade de redundância como: imagens, vídeos e áudios. Existem também trabalhos que buscam ocultar informações utilizando esteganografia em arquivos com pouca redundância, como textos. Há ainda, alguns trabalhos que buscam utilizar a lógica de jogos para ocultar dados digitais por meio da esteganografia (80).

4.2 Esteganografia Digital em Imagens

É possível utilizar técnicas de esteganografia para armazenar informações em imagens digitais que são representadas no formato matricial. Tais técnicas aplicadas em imagens são comumente divididas entre técnicas de esteganografia: *i)* com incorporação, ou *ii)* sem incorporação. As técnicas de esteganografia com incorporação permitem que o usuário escolha a imagem que será utilizada para conter o arquivo secreto, enquanto que nas técnicas de esteganografia sem incorporação a imagem que conterá a informação oculta é gerada a partir do arquivo secreto, portanto, não é possível que o usuário a escolha. É importante ressaltar que a imagem gerada a partir das técnicas de esteganografia digital sem incorporação conterá significado visual para o ser humano como uma imagem de uma textura ou gerada por uma GAN, e o seu significado difere da informação contida no arquivo escondido (81, 20, 82).

As técnicas de esteganografia com incorporação ainda podem ser subdivididas em domínio: *i)* espacial, ou *ii)* da frequência. Enquanto técnicas no domínio espacial normalmente apresentam melhores resultados quando se considera imagens armazenadas nos formatos de arquivos sem compressão ou com compressão *lossless* como BMP, PNG ou TIFF, as técnicas no domínio da frequência são baseadas em algumas transformadas e apresentam melhores resultados em formatos que utilizam as respectivas transformadas no seu processo de codificação. Por exemplo, a técnica de esteganografia digital baseada na transformada do cosseno discreto apresenta melhores resultados em imagens armazenadas no formato JPEG (83). A seguir será abordado, sucintamente, o domínio das técnicas de esteganografia supracitados.

4.2.1 Domínio Espacial

As técnicas de esteganografia no domínio espacial no mais das vezes consistem em armazenar a mensagem nos *bits* menos significativos de cada canal de cada pixel de uma imagem digital. Nestas técnicas os *bits* selecionados para conter a mensagem secreta são substituídos pelos *bits* da mensagem secreta. A ordem pela qual os *bits* serão processados é definida pela técnica a ser utilizada. Visando dificultar ainda mais a detecção e extração da mensagem secreta, a imagem pode ser convertida para outro espaço de cores antes da aplicação da técnica de esteganografia. Ao concluir a aplicação da esteganografia, a imagem é convertida novamente para seu formato original de modo a ser salva. No caso de se utilizar conversão no espaço de cores vale evidenciar que é preciso escolher os espaços de cores utilizados, de modo a não haver alterações nos *bits* contendo a mensagem secreta durante o processo de conversão. Tal cuidado é indispensável, pois para recuperar a mensagem secreta é necessário converter novamente a imagem para o espaço de cor previamente definido, com o intuito de depois extrair a mensagem secreta dos *bits* pré-determinados. A seguir, serão descritas algumas técnicas de esteganografia no domínio espacial.

1. ***Bit* Menos Significativo**

As abordagens mais comuns para realização de inserção de mensagens secretas em imagens usando ruído são baseadas na técnica LSB (*Least Significant Bit*), sendo este 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 15 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 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, seguramente, sem causar alterações que sejam perceptíveis visualmente na mesma (84, 85). Neste trabalho, 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 (86). Portanto, este trabalho 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 15.

Figura 15: Representação de cada pixel da imagem utilizando a técnica LSB.

	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

Fonte: Próprio autor

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 implementada.

O Algoritmo 1 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 $getSizes(i_{bits})$ (linha 2). A partir da quantidade de *bits* da imagem de capa calcula-se o tamanho máximo que a mensagem secreta pode ter, para ser possível escondê-la na 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 na 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

Algoritmo 1: LSB (i_c, msg)

1: $i_{bits} \leftarrow$ getBits(i_c)
2: $size_{i_{bits}} \leftarrow$ getSize(i_{bits})
3: $tamMaxMsg \leftarrow \lfloor size_{i_{bits}}/4 \rfloor$
4: **if** getSize(msg) $\leq tamMaxMsg$ **then**
5: $bitsMsg[\,] \leftarrow$ getAsciiBits(msg)
6: $contadorBits \leftarrow 7$
7: **for** each $\{bit_1, bit_2\} \in bitsMsg$ **do**
8: $i_{bits}[contadorBits] \leftarrow bit_1$
9: $i_{bits}[contadorBits + 1] \leftarrow bit_2$
10: $contadorBits$ += 8
11: **end for**
12: $i_m \leftarrow$ bitsToImage(i_{bits})
13: **return** i_m
14: **end if**

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 termina retornando a imagem recebida, agora, 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.

2. ***Bit* Menos Significativo em Escala de Cinza**

A técnica LSB em escala de cinza realiza o processo equivalente ao método LSB em imagens RGB, porém utilizando escala de cinza. Neste trabalho convertemos as imagens do conjunto de testes para escala de cinza, a fim de avaliar o desempenho esta técnica. A utilização de imagens em escala de cinza visa reduzir tanto as distorções visuais na imagem pós-processada, quanto melhorar os valores obtidos nas métricas de avaliação utilizadas, as quais serão abordadas adiante: *(i) Mean Squared Error* (MSE) e *(ii) Peak Signal to Noise Ratio* (PSNR).

Esta técnica é ilustrada na Figura 16, onde cada linha reflete um pixel da matriz que representa a 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 permanecem inalterados no final do procedimento e os que estão caracterizados pela cor cinza representam os que foram alterados ao armazenar a mensagem secreta. Neste trabalho, 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 à recomendação *ITU-R Recommendation BT.709* (87).

3. **4º *Bit* em Escala de Cinza (SSB-4)**

Em algumas das técnicas de esteganografia digital presentes na literatura são utilizados os *bits*

Figura 16: Representação de cada pixel da imagem utilizando a técnica LSB 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

Fonte: Próprio autor

menos significativos de uma imagem para armazenar a mensagem secreta. Contudo, visando dificultar a 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 (88).

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 (20):

$$n = \lfloor getSize(msg) \rfloor , \tag{20}$$

onde *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, que consiste em alterar os valores destes *bits* de modo 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 exemplificada na Figura 17, em que 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* destacados em cinza são alterados para incorporar a mensagem secreta. No que lhes concerne, 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.

4. ***Bit* Aleatório em Escala de Cinza (SSB-N)**

A técnica SSB-N, abordagem proposta neste trabalho e que tem como inspiração a técnica SSB-4, 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

Figura 17: Representação de cada pixel da imagem utilizando a técnica SSB-4.

	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

Fonte: Próprio autor

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 (i.e., que conterá a mensagem secreta) ocorre de forma 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 dada esta escolha aleatória, se faz 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 (20), e o tamanho de n em *bytes* é calculado através da Equação (21):

$$n_{bytes} = \left\lfloor \frac{getSize(i_c)/8}{getSize(msg)} \right\rfloor, \tag{21}$$

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.

A técnica SSB-N é ilustrada através da Figura 18, onde cada linha representa um pixel da matriz da imagem sendo percorrida da esquerda para a direita, linha a linha (89).

Figura 18: Representação de cada pixel da imagem utilizando a técnica SSB-N.

	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

Fonte: Próprio autor

4.2.2 Domínio da Frequência

Técnicas de esteganografia no domínio da frequência utilizam os coeficientes resultantes de alguma transformada para esconder a mensagem secreta. Estas técnicas são comumente aplicadas nos formatos de arquivos que utilizam a respectiva transformada para codificar a imagem. Por exemplo, as técnicas de esteganografia em imagens baseadas na DCT (transformada do cosseno discreto, do inglês, *Discrete Cosine Transform*) são geralmente utilizadas para ocultar arquivos em imagens no formato JPEG. A seguir será detalhada uma técnica que utiliza do *bit* menos significativo do coeficiente da DCT. Visando efetuar uma comparação será apresentada uma técnica que se utiliza do *bit* menos significativo do coeficiente da FFT (Transformada Rápida de Fourier, do inglês, *Fast Fourrier Transform*) para ocultar os *bits* do arquivo secreto.

1. **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 discreto, já presente no algoritmo do respectivo formato da imagem, que converte blocos sucessivos de tamanho 8×8 pixels em 64 coeficientes de uma DCT. Portanto, os *bits* menos significativos dos coeficientes das DCTs 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 (19). Os *bits* do arquivo secreto podem ser armazenados alterando-se os *bits* menos significativos dos coeficientes de baixa frequência de cada bloco 8×8 de uma imagem JPEG. É importante ressaltar que essa alteração deve ser executada após a divisão do bloco pela matriz de quantização no processo de codificação de uma imagem JPEG, para evitar que a mensagem oculta se perca.

   Como apresentado em (90) a transformada do cosseno discreto 2D é definida da seguinte forma (Equações (22), (23) e (24)):

$$F(u,v) = C_u C_v \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2M}\right), \tag{22}$$

$$C_u = \begin{cases} \frac{1}{\sqrt{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1,2,...,N-1 \end{cases}, \tag{23}$$

$$C_v = \begin{cases} \frac{1}{\sqrt{M}} & v = 0 \\ \sqrt{\frac{2}{M}} & v = 1,2,...,M-1 \end{cases}, \tag{24}$$

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 (23) e (24), onde N e M representam as dimensões da imagem, largura e altura, respectivamente.

2. **Transformada Rápida de *Fourier***

   A técnica de esteganografia baseada na transformada de *Fourier* é muito semelhante à baseada na DCT. Todavia, a FFT implementa de maneira computacionalmente eficiente a transformação

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y)\, exp\left(-j\frac{2\pi(ux+vy)}{N}\right), \tag{25}$$

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

   Assim como na técnica baseada em 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.

4.3 Esteganálise

Esteganálise consiste no conjunto de técnicas que permitem detectar arquivos ocultos em outros arquivos utilizando alguma técnica de esteganografia digital. Ao contrário da criptoanálise, que para ser bem sucedida precisa quebrar a criptografia, ou seja, decodificar a mensagem codificada sem ter conhecimento da chave criptográfica necessária para decodificar determinada mensagem, na esteganálise, basta detectar a existência de um dado escondido em um arquivo digital para ser considerado bem-sucedido. Todavia, embora não seja necessário para o procedimento ser considerado

bem-sucedido, a revelação da mensagem secreta pode ser executada a partir da esteganálise se possível (91).

As principais técnicas de esteganálise se dividem em seis categorias nomeadas como: visual, de assinatura ou específica, estatística, de espectro espalhado, no domínio das transformadas e universal ou cega. A seguir serão brevemente descritas as principais características das categorias supramencionadas (91).

1. **Esteganálise Visual**

   Consiste no examinador observar cuidadosamente a imagem duvidosa visando detectar alterações suspeitas que possam ser indícios de que a imagem contenha um arquivo escondido utilizando esteganografia. Na esteganálise visual também podem ser aplicados alguns filtros na imagem para serem exibidos apenas os valores dos *bits* menos significativos, ou seja, a região onde é geralmente armazenado o arquivo secreto (91).

2. **Esteganálise de Assinatura ou Específica**

   Busca por assinaturas que *softwares* de aplicação de esteganografia deixam nos arquivos processados por eles. Por exemplo, o *software Masker* utiliza os últimos 77 *bytes* da imagem para armazenar sua assinatura. Já o *software* Jpegx adiciona sua assinatura antes do marcador de final de arquivo especificado na documentação do formato JFIF que é o padrão utilizado em arquivos Jpeg. A assinatura do *software* Jpegx consiste na seguinte sequência de *bytes*: 5B 3B 31 53 00. Cada *software* de aplicação de esteganografia possui sua assinatura correspondente (91).

3. **Esteganálise Estatística**

   Consiste em analisar a técnica de esteganografia que se suspeita ter sido utilizada e identificar quais as medidas estatísticas na imagem que a respectiva técnica deveria alterar. Após este procedimento o examinador obtém estas medidas estatísticas para a imagem suspeita, e a partir da análise destes valores deve determinar se a imagem contém ou não esteganografia (91).

4. **Esteganálise de Espectro Espalhado**

   A esteganografia de espectro espalhado consiste em representar o arquivo secreto através de um ruído, e adicionar este ruído na imagem que conterá o arquivo secreto. É importante destacar que este ruído é invisível para os seres humanos. Para detecção desta categoria de esteganografia foram propostas as técnicas de esteganálise de espectro espalhado, sendo que a principal delas utiliza transformada de Fourier do histograma da imagem para detectar a esteganografia (91).

5. **Esteganálise no Domínio das Transformadas**

   Visa detectar esteganografia no domínio da frequência. Para isto a imagem é dividida em blocos de 8×8, é calculada a transformada do cosseno discreto e a transformada *wavelet* de cada bloco, em seguida são mensuradas algumas propriedades estatísticas dos coeficientes de cada

transformada em cada bloco. Estas propriedades são fornecidas para uma rede neural artificial que classifica se a imagem suspeita contém esteganografia ou não (91).

A transformada *wavelet* baseia-se na definição de uma *wavelet*. Uma *wavelet* é a representação de uma oscilação ondulatória localizada no tempo. Esta oscilação tem duas propriedades: *i)* escala, e *ii)* localização. A escala consiste em definir quão esticada ou esmagada é uma *wavelet*, enquanto a localização define onde a *wavelet* está localizada no tempo. Um exemplo de uma *wavelet* pode ser definida pela Equação (26), na qual o parâmetro a representa a escala e o parâmetro b a localização.

$$f(x) = -(x-b)e^{\frac{-(x-b)^2/(2a^2)}{\sqrt{2\pi a^3}}} \tag{26}$$

A transformada *wavelet* consiste em uma convolução de um sinal com um conjunto de *wavelets* em uma variedade de escalas. Com isso será calculado o quanto de uma *wavelet* está em um sinal para determinada escala e localização. Existem duas categorias de transformada wavelets: *i)* contínua e *ii)* discreta. A contínua representada na Equação (27), onde a representa a escala da *wavelet*, b representa a localização da wavelet, t representa o tempo, $*$ o símbolo do complexo conjugado e ψ uma função *wavelet*. A transformada discreta é representada na Equação (28), onde m é um conjunto finito de escalas, n um conjunto finito de localizações e ψ uma função *wavelet*.

$$T(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi^* \frac{(t-b)}{a} dt \tag{27}$$

$$T_{m,n} = \int_{-\infty}^{\infty} x(t)\psi_{m,n}(t)dt \tag{28}$$

6. **Esteganálise Universal ou Cega**

A esteganálise universal ou cega detecta esteganografia em imagens independentemente da classe de técnicas de esteganografia utilizada para esconder a mensagem secreta nas imagens. As ferramentas mais recentes de esteganálise universal ou cega utilizam redes neurais convolucionais profundas para extrair características da imagem, e a partir destas características classificar se a imagem contém esteganografia ou não. É importante ressaltar que a identificação da esteganografia em imagens sem conhecimento da técnica utilizada não é uma tarefa trivial, dada a abundância de técnicas disponíveis na literatura e a infinidade de imagens diferentes que podem existir. Estas técnicas baseadas em redes neurais convolucionais profundas geralmente apresentam um bom resultado quando treinadas no mesmo conjunto de imagens utilizado posteriormente para armazenar os arquivos ocultos. Todavia, quando utilizadas no mundo real, onde na maioria das vezes não se tem acesso ao conjunto original de imagens, estas técnicas não

apresentam um bom resultado (91).

4.4 Comentários finais

Após a exposição dos conceitos de esteganografia e de esteganálise feita neste capítulo, trataremos dos trabalhos relacionados a estes tópicos no próximo capítulo.

5 Trabalhos Relacionados

Existem na literatura variados trabalhos que visam armazenar arquivos digitais dentro de outros arquivos digitais utilizando diversas técnicas de esteganografia digital em diferentes categorias de ficheiros. A seguir serão apresentados vários trabalhos que propõem técnicas de esteganografia digital para imagens, vídeos, áudio e jogos. Estes trabalhos utilizam classes distintas de esteganografia, por exemplo, para imagens podemos citar, esteganografia: *i)* domínio espacial, *ii)* domínio da frequência, *iii)* com incorporação, ou *iv)* sem incorporação.

Os trabalhos apresentados neste capítulo foram obtidos em uma revisão sistemática da literatura. Estes artigos foram mapeados através do banco de dados sobre periódicos *Science Direct* e também por pesquisas no Google Acadêmico. Para conduzir a busca na plataforma foi utilizada a *string* de busca *"Digital Steganography AND image processing AND NOT (video OR audio OR voice OR bio OR steganalysis OR watermark)" e "generative steganography AND image processing AND NOT audio"*, limitando-se a pequisa aos últimos cinco anos. Foram utilizadas duas chaves de buscas diferentes, pois a primeira tem o objetivo de pesquisar por trabalhos relacionados às técnicas de esteganografia com incorporação, enquanto a segunda visa buscar trabalhos de esteganografia sem incorporação. É importante ressaltar que a primeira chave de busca apresenta mais critérios de exclusão em comparação com a segunda, pois as técnicas de esteganografia com incorporação são amplamente utilizadas em diversos tipos de arquivos redundantes, como imagens, áudios, vídeos, entre outros. Já as técnicas de esteganografia sem incorporação são mais recentes e até o momento são utilizadas principalmente em imagens. Dada a limitações no tamanho da *string* da chave de busca imposta pela plataforma foi inviabilizado utilizar apenas uma chave de busca para ambos os temas.

A primeira chave de busca retornou 91 resultados, enquanto a segunda chave, por constituir um campo mais recente, retornou 35 resultados, totalizando 126 trabalhos obtidos. A partir da análise do *abstract* e introdução destes trabalhos foram aplicados critérios de inclusão e exclusão visando selecionar os que apresentam mais correspondência com o trabalho proposto. Os critérios de inclusão considerados são, trabalhos: *i)* que tratam de esteganografia em imagens digitais, *ii)* que abordam mecanismos de marca d'água digital utilizando ou baseando-se em técnicas de esteganografia digital, *iii)* sobre computação forense que abordam técnicas de esteganografia. Os critérios de exclusão utilizados consistiam em trabalhos: *i)* que abordam apenas técnicas de criptografia, *ii)* sobre bio-esteganografia, *iii)* sobre marca d'água digital que não tenham relação com técnicas de esteganografia, *iv)* sobre esteganografia em áudio ou vídeo. Foram selecionados 28 trabalhos, apresentados na Tabela 1. Estes foram completamente analisados e serão descritos a seguir.

O trabalho (92) visa analisar, detectar e mitigar ataques de port-scan, entre outros ataques de redes, utilizando uma rede neural convolucional profunda. A abordagem proposta apresentou uma precisão de 92% no conjunto de treinamento e 61% nos dados de teste. Já o trabalho (93) visa encontrar uma relação equilibrada entre qualidade da imagem, capacidade de armazenamento e segurança do método de esteganografia. Para isto uma nova abordagem de esteganografia é proposta baseada

Tabela 1: Classificação dos Trabalhos Relacionados por Categoria de Técnicas de Esteganografia

Referência	C/ Incorporação		S/ Incorporação		Outras Técnicas		
	D. Espacial	D. Frequência	G. Textura	GAN	Criptografia	Compressão	CNN
(92)							✓
(93)	✓				✓	✓	
(94)	✓	✓					
(95)	✓	✓					
(96)	✓						
(97)							✓
(98)	✓	✓			✓		
(99)	✓				✓		
(100)[4]							
(101)					✓		
(102)	✓	✓			✓		
(103)	✓						
(104)	✓						
(105)					✓		
(106)	✓				✓		
(107)		✓					
(108)	✓						
(109)					✓		
(110)					✓		
(111)							✓
(112)							✓
(113)				✓			✓
(114)							
(115)				✓			✓
(116)				✓	✓		✓
(117)							✓
(118)							✓
(119)							✓

no algoritmo de criptografia Vigenere Cipher e o algoritmo de compressão Huffman Coding. Os resultados empíricos demonstraram que o algoritmo proposto é superior aos algoritmos mais antigos quando avaliado em relação à qualidade da imagem pós-processada, PSNR de 55,71dB, e capacidade de armazenamento, 52.400 *bytes*.

O trabalho (94) apresenta dois objetivos principais, sendo: *i)* apresentar as bases de dados para computação forense disponíveis na literatura e na *internet*, e *ii)* enfatizar a importância para que pesquisadores disponibilizem suas bases de dados permitindo a replicação de resultados e melhorando o estado da arte. O autor avaliou 715 artigos extraídos da literatura da área, chegando à conclusão que existem muitos conjuntos de dados para uso, porém encontrá-los é um desafio. Já o trabalho (95) analisa diversas técnicas utilizadas por *malwares* para persistir no dispositivo invadido e infectar outros

dispositivos, uma das técnicas utilizadas e analisadas consiste na esteganografia digital. As técnicas analisadas visam permitir ao *malware* se comportar como um componente legítimo do sistema de destino.

O trabalho (96) visa encontrar uma relação equilibrada entre distorção visual e capacidade de carga do sistema de esteganografia. Para isto, propõe uma técnica de esteganografia baseada em algoritmos genéticos. Os dados secretos são armazenados nos *bits* menos significativos da imagem. Entretanto, antes de serem armazenados os dados secretos são pré-processados pelo algoritmo proposto, que utiliza alguns parâmetros definidos a partir de algoritmo genético. Os resultados apresentam um valor médio de PSNR de 46,41 dB e 40,83 dB para 2 *bits* por pixel (bpp) e capacidade de ocultação de dados de 3 bpp. Já o trabalho (97) consiste em uma revisão da literatura que discute sobre a aplicação de algoritmos de processamento de imagens aplicados a problemas da área biomédica.

Já o trabalho (98) visa proteger a transmissão imagens digitais na rede mundial de computadores. Para isto o autor propõe um método composto pela utilização de criptografia e esteganografia. O método proposto é baseado em uma nova categoria de mapa de caos híbrido 2D distribuído uniformemente baseado em mapas logísticos, mapa de seno e mapa *Tent*. No método proposto também são utilizados autômatos celulares e a transformada discreta de Framelet, além de aplicar deslocamentos para misturar a posição dos píxeis da imagem. Foram realizadas diversas simulações de ataques e análises de segurança para demonstrar a eficácia do método proposto.

O trabalho (99) propõe um algoritmo para criptografia rápida e segura de imagens em lote. Dado o crescimento da quantidade de imagens trafegadas diariamente na *internet*, criptografar grandes bases de imagens se tornou um problema recorrente e relevante. O método proposto utiliza criptografia baseada em propriedades intrínsecas da imagem, esteganografia reversível para armazenamento de metadados na imagem criptografada e computação paralela. Os resultados experimentais e análises de segurança demonstram que o método proposto pode alcançar resultados superiores com alta eficiência. O trabalho (100) baseia-se em um método de esteganografia em redes de computadores, ou seja, a mensagem secreta é ocultada nos pacotes que trafegam na rede. É proposto um método de esteganografia sem perdas, que incorpora dados secretos em *strings* de dados codificados. Os resultados dos experimentos realizados demonstram o funcionamento adequado da técnica proposta.

O trabalho (101) visa resolver problemas onde os algoritmos de criptografia de imagens transformam a imagem original em uma imagem contendo apenas ruído. Isto facilita a identificação acerca da suposta utilização de criptografia em uma determinada imagem. O autor propõe um algoritmo de criptografia de imagens sem perdas baseado no algoritmo de Bao & Zhou. Tal algoritmo gera uma imagem cifrada, visualmente significativa que dificulta a identificação de um processo de criptografia na imagem. Resultados empíricos são apresentados para evidenciar a alta qualidade das imagens resultantes e o desempenho do algoritmo proposto. Um novo esquema de compartilhamento seguro de dados baseado em *blockchain* utilizando esteganografia de imagem e técnicas de criptografia para aplicações de telemedicina é proposto em (102). Foi realizada uma extensa análise experimental e os resultados das simulações indicaram que o algoritmo proposto obteve uma relação sinal ruído de

pico (PSNR) de 51.75 dB. O trabalho (103) introduz a arquitetura para esteganografia de imagens baseadas em autômatos celulares de pontos quânticos. Foi proposto um circuito de esteganografia codificador/decodificador baseado na esteganografia do *bit* menos significativo. Para o projeto do circuito foi desenvolvida uma nova porta QCA XOR que apresenta baixa densidade de dispositivos e menor contagem de células em comparação a outros circuitos existentes. Foram realizadas simulações e experimentos para análise de potência do circuito proposto, sendo os resultados aceitáveis.

Um método de esteganografia baseado em interpolação de imagens é proposto em (104), visando armazenar ocultamente uma imagem dentro de outra imagem. Para isto é utilizada a interpolação média de vizinhança e substituição de *bits* menos significativos para armazenar a imagem oculta. Além disso, é executado um processo de ajuste de pixel ideal para melhorar a qualidade visual da imagem contendo esteganografia. O principal resultado deste artigo consiste em uma prova teórica de que o esquema proposto apresenta melhores resultados quando comparado com o esquema de Jung & Yoo. Em (105) é feito um estudo de técnicas de criptografia visual dinâmica nas quais é necessário um computador para criptografar a imagem, sem a necessidade de um computador para descriptografar a mensagem. Para descriptografia são utilizados instrumentos ópticos ou em alguns casos utiliza-se apenas o sistema visual humano. Além disso, é proposto um novo esquema para criptografia visual dinâmica que combina padrão Moiré e matriz de píxeis. Foram realizados simulações que mostram que a abordagem proposta pode atender a demanda de segurança de forma satisfatória.

O trabalho (106) avalia a utilização da esteganografia do *bit* menos significativo para armazenamento de texto em imagem visando garantir a segurança da informação ao ser trafegada pela rede. Além de esconder o texto, antes do mesmo ser armazenado na imagem, é realizada a criptografia do texto utilizando o esquema criptográfico McEliece por código Goppa. Os experimentos realizados demonstraram que o método proposto consegue proteger informações de atacantes de maneira satisfatória. Já o trabalho (107) apresenta um novo mecanismo de imagem correlacionada de pixel único, que pode ser utilizado para autenticação de várias imagens. Nos experimentos realizados observou-se resultados satisfatórios para autenticação das imagens.

O trabalho (108) consiste em um *survey* de métodos que podem ser utilizados para proteger carros conectados em seu procedimento de atualização de *software*. É uma tendência em todo mundo a utilização de atualizações de *software over-the-air* em veículos conectados. Este mecanismo de atualização, embora solucione a atualização do *software* de veículos que se encontram em regiões onde o acesso à *internet* é limitado, também cria algumas lacunas de segurança no sistema, que precisam ser tratadas. O *survey* citado tem como resultado a apresentação de diversas perspectivas para implantação de atualizações de *software over-the-air*, bem como utilização de esteganografia para aumentar a segurança. Já o trabalho (109) visa melhorar a qualidade visual de imagens criptografadas, dado que as mesmas normalmente se assemelham a imagens geradas aleatoriamente, portanto, outro usuário pode suspeitar que determinada imagem foi criptografada. Os experimentos realizados demonstraram resultados satisfatórios.

Em (110) são apresentados algoritmos de criptografia de imagens para melhorar a qualidade

visual de imagens pós-processadas criptografadas. Tais algoritmos garantem que a imagem resultante tenha um significado visual e não seja apenas uma imagem randômica. Para isto é proposto um algoritmo onde a imagem é pré-criptografada, e em seguida é incorporada nas sub-bandas da transformada wavelet inteira da imagem de capa. Os experimentos realizados mostraram que a qualidade visual das imagens resultantes do método proposto é maior que a qualidade visual das imagens processadas pelos demais métodos existentes. Em (111) é avaliada a utilização de métodos de aprendizado profundo, como redes neurais convolucionais, para avaliar imagens de radiografia de Tórax e tomografia computadorizada visando auxiliar na detecção de covid-19. Visando auxiliar trabalhos futuros foram realizados experimentos onde foram coletados algumas métricas de avaliação para as técnicas em análise.

O trabalho (112) é um *survey* que revisa as técnicas mais recentes baseadas em aprendizado profundo e visão computacional para análise forense de vídeos. Discute a detecção de: *i) deep fakes*, *ii)* edição de cenas em vídeos, *iii)* mensagens armazenadas em vídeos utilizando técnicas de esteganografia digital. Além disso os principais desafios enfrentados por pesquisadores de análise forense de vídeos são discutidos. Em (113) são propostas três arquiteturas de redes neurais adversárias generativas para aplicação de esteganografia a partir do jogo entre três jogadores. O jogo de três jogadores é um conceito recorrente em trabalhos de esteganografia envolvendo redes neurais adversárias generativas. Consiste em um jogo onde três jogadores competem entre si, sendo que cada jogador tem um objetivo. O primeiro jogador gera uma imagem contendo significado visual e uma mensagem secreta, o segundo revela a mensagem escondida na imagem, enquanto o terceiro identifica se existe comunicação oculta na imagem. À medida que o jogo avança é esperado que o terceiro jogador não consiga identificar se existe mensagem na imagem, enquanto o primeiro e segundo jogares conseguem cumprir seus objetivos com eficiência. O método proposto alcançou melhores resultados em relação aos métodos existentes.

O trabalho (114) apresenta um estudo sobre métodos para garantir a privacidade dos usuários de *softwares* de comunicação como redes sociais, correio eletrônico, entre outros. Entre os conjuntos de métodos analisados encontra-se a esteganografia digital. É importante ressaltar que é apresentada apenas uma análise teórica no estudo. Em (115) é apresentado um novo método de esteganografia baseado em redes neurais adversárias generativas visando melhorar a segurança da esteganografia quando atacada por redes neurais de esteganálise. O método proposto é denominado SPS-ENH, do inglês, *SParSe ENHancement*. Foram realizados diversos experimentos que demonstraram que o método proposto é superior aos demais métodos existentes, quando avaliado em relação à segurança da mensagem armazenada na imagem. Ou seja, a acurácia das redes neurais de esteganálise é menor quando aplicadas a imagens processadas por este método. O trabalho (116) propõe a utilização de redes neurais adversárias generativas na tarefa de criptografar uma imagem gerando outra imagem com significado visual. A primeira rede tem o objetivo de gerar uma imagem sintética contendo significado visual. Já a segunda irá armazenar a imagem secreta na imagem de capa gerada pela primeira rede. Por fim, a terceira tem o objetivo de reconstruir a imagem escondida pela segunda a partir da imagem resultante

do processamento da segunda rede. Foram realizados diversos experimentos utilizando-se imagens extraídas do *dataset* MNIST. A robustez do método proposto em relação à aplicação de ruídos na imagem contendo a mensagem secreta foi avaliada. Métricas de qualidade das imagens pós-processadas foram utilizadas no processo de avaliação. Os resultados apresentados foram satisfatórios.

O trabalho (117) apresenta diversas abordagens para esteganálise utilizando aprendizado profundo propostas entre 2015 a 2018. Foi realizada uma análise teórica das redes neurais apresentadas, sendo o resultado discutido no trabalho. Um novo método para destruir dados armazenados em imagens através de esteganografia em redes sociais é proposto em (118). Uma rede neural é utilizada para processar a imagem contendo esteganografia a fim de retornar uma imagem limpa, ou seja, uma imagem visualmente semelhante à inicial, porém, que não tenha dados escondidos. Diversos experimentos foram realizados, onde se nota que o método proposto funciona corretamente para remover diversos mecanismos diferentes de esteganografia. Além disso, a imagem pós-processada é de boa qualidade e o processamento é rápido.

Visando aprimorar a segurança do sistema de esteganografia, em (119) é proposto um esquema de esteganografia baseado em transferência de estilo. Foram realizados diversos experimentos que indicaram que o sistema proposto consegue extrair uma mensagem oculta de uma imagem com uma baixa taxa de erro de *bit*. Além disso, o sistema proposto oferece muita segurança.

Além dos trabalhos analisados durante a revisão sistemática da literatura, foram examinados outros artigos durante o processo de implementação das técnicas. Estes trabalhos forneceram o arcabouço técnico necessário para a implementação das técnicas que estão sendo comparadas, além de fornecer uma inspiração para proposta das técnicas: *i)* sistema de esteganografia do *n*-ésimo *bit* e *ii)* esteganografia a partir da utilização de autoencoders. Estes trabalhos foram encontrados por buscas na plataforma Google Acadêmico e serão apresentados a seguir.

O trabalho (20) propõe uma técnica de esteganografia sem incorporação em imagens onde são utilizadas redes neurais convolucionais profundas. A mensagem secreta é escondida na semente de uma rede geradora que irá gerar uma imagem contendo um significado visual para o ser humano enquanto representa ocultamente a mensagem secreta. Para revelar a mensagem secreta é utilizada uma rede neural extratora. A mensagem revelada pela rede extratora pode conter alguns erros. Outro trabalho com uma abordagem semelhante é proposto em (120).

O artigo (121) propõe uma técnica de esteganografia digital com incorporação baseada em redes neurais convolucionais. São utilizadas três redes neurais convolucionais. A primeira delas tem a função de esconder dados dentro de uma imagem. Já a segunda tem a função de revelar os dados escondidos. Por fim, a terceira é uma rede neural de esteganálise que visa detectar se a imagem contém ou não esteganografia. As três redes são treinadas simultaneamente, de modo que a primeira aprenda como esconder um arquivo em uma imagem de forma que a segunda possa revelar o arquivo enquanto engane a terceira. A segunda irá aprender a revelar o arquivo oculto na imagem. A terceira consiste em uma rede neural de esteganálise padrão pré treinada que irá tentar identificar a existência ou não de esteganografia na imagem.

Em (122) é proposta uma técnica que combina redes neurais adversárias generativas e esteganografia no domínio espacial para implementar um sistema de esteganografia em imagens que torne a tarefa de esteganálise mais difícil. Com isso, espera-se reduzir os riscos da detecção da mensagem secreta. Uma forma de otimizar a técnica de esteganografia baseada no jogo entre Alice, Eve & Bob é apresentada em (123).

O trabalho (124) propõe um método para esteganografia digital em imagens JPEG que consiste em armazenar a mensagem secreta nos coeficientes da transformada cosseno discreto de cada bloco 8×8 da imagem durante o processo de codificação JPEG. É importante ressaltar que neste trabalho as alterações são feitas nos coeficientes da DCT de modo a simular na imagem a mesma distorção gerada pela alteração da configuração ISO de uma câmera digital. Com isso é esperado que o examinador julgue que a distorção foi gerada pelas configurações da câmera e não por uma técnica de esteganografia digital. No trabalho (125) é proposta uma técnica de esteganografia para vídeos utilizando a codificação H.264. Esta técnica utiliza uma variação do algoritmo de codificação de canal código treliça progressiva para armazenar e recuperar as mensagens secretas.

O artigo (126) utiliza redes neurais convolucionais na tarefa de esteganografia digital em arquivos de áudio. Este artigo apresenta muitos detalhes específicos da esteganografia em arquivos de áudio no domínio do tempo, além de apresentar diversos trabalhos relacionados a esteganografia em arquivos de áudio. Em (127) é proposto um método de esteganografia em jogos. O método baseia-se em utilizar a posição das bombas em um tabuleiro do jogo campo minado para esconder a mensagem secreta. Considerando que o jogo de campo minado está sendo acessado através da *internet*, é possível esconder a transmissão de uma mensagem conforme o tabuleiro que será transmitido pelo servidor do jogo ao cliente. Este trabalho também apresenta uma análise sobre a segurança deste método de esteganografia.

Em (128) é proposto um método para ocultar uma imagem em escala de cinza dentro de uma imagem RGB utilizando uma CNN denominada *encoder*. Para revelar a imagem em escala de cinza a partir da RGB pós-processada é utilizado outra CNN denominada *decoder*. O trabalho (129) também utiliza uma CNN para esconder uma imagem dentro de outra, sendo utilizada outra rede convolucional para recuperar a que foi escondida a partir da imagem pós-processada. A diferença deste trabalho para o anterior consiste no fato que neste ambas as imagens, pública e oculta, são RGB, enquanto que no trabalho anterior oculta-se uma imagem em escala de cinza em uma RGB.

Alguns trabalhos presentes na literatura utilizam redes neurais convolucionais em topologias como de um autoencoder para armazenar e recuperar arquivos de imagens em arquivos de imagens, sendo que tanto a imagem escondida quanto a imagem pública sofrem distorções durante este processamento (128, 129). Tendo em vista esta varredura bibliográfica, cabe ressaltar que o que difere este trabalho dos demais consiste na abordagem utilizada para armazenar arquivos binários em arquivos de imagens, utilizando redes neurais convolucionais com topologia como de um autoencoder de modo que seja possível recuperar o arquivo binário sem alterações, na maioria dos casos. Adicionalmente, neste trabalho é apresentada uma comparação entre algumas técnicas extraídas da literatura. É impor-

tante ressaltar que para esta comparação as técnicas foram executadas no mesmo conjunto de imagens utilizado nos experimentos dos métodos propostos.

6 Metodologia

A metodologia utilizada para escolha, implementação e validação das técnicas de esteganografia descritas neste trabalho é apresentada neste capítulo. Nas seções 6.1, 6.2 e 6.3 são apresentadas a escolha das técnicas de esteganografia, ferramentas de esteganálise e métricas de qualidade, respectivamente. Já na seção 6.4 é descrito como os *datasets* de imagens foram obtidos. O procedimento de implementação das técnicas de esteganografia analisadas neste trabalho é apresentado na seção 6.5.

6.1 Seleção das Técnicas de Esteganografia

As técnicas de esteganografia que compõem este trabalho foram selecionadas visando abranger métodos de aplicação da esteganografia que apresentem características disjuntas entre si, consequentemente abrangendo técnicas de diferentes domínios e classificações, em simultâneo, em que fosse possível realizar uma comparação justa dos métodos analisados.

Após uma pesquisa na literatura da área, visando obter técnicas para aplicação de esteganografia em imagens, optou-se por selecionar as seguintes técnicas no domínio espacial: *i)* Sistema de esteganografia do *bit* menos significativo; *ii)* Sistema de esteganografia do *bit* 4; *iii)* Sistema de esteganografia do *bit N*, proposta neste trabalho inspirada na técnica *ii)*.

Entretanto, visando abranger também técnicas no domínio da frequência, foram selecionadas técnicas de esteganografia baseadas em transformada do cosseno discreto, a partir da literatura da área. Além das técnicas anteriormente citadas, tendo o objetivo de explorar a concepção de uma técnica de esteganografia com características distintas das técnicas atualmente disponíveis na literatura, é avaliada neste trabalho a utilização de *denoising autoencoders* para aplicação de esteganografia em imagens.

6.2 Seleção da Ferramenta de Esteganálise

Após uma pesquisa por ferramentas para realizar esteganálise, de código aberto, capaz de suportar os formatos de imagens processados neste trabalho, optou-se por utilizar uma ferramenta de esteganálise denominada *StegExpose*. Esta é uma ferramenta de código aberto que emprega diversas técnicas de esteganálise visando investigar propriedades da imagem mirando detectar esteganografia. O código-fonte da ferramenta *StegExpose* encontra-se publicado em um repositório aberto no GitHub[5]. Entre as técnicas de esteganálise disponíveis nesta ferramenta podemos citar: *i) Sample Pairs by Dumitrescu* (2003); *ii) RS Analysis by Fridrich* (2001); *iii) Chi Square Attack by Westfeld* (2000); *iv) Primary Sets by Dumitrescu* (2002) (130). Estas técnicas não são apresentadas em detalhes neste trabalho, pois se encontram fora do escopo do mesmo. O foco deste trabalho refere-se às técnicas de esteganografia, sendo as técnicas de esteganálise utilizadas como uma ferramenta para validar

[5] https://github.com/b3dk7/StegExpose

a qualidade e segurança da esteganografia.

Discutiu-se e foram realizados experimentos preliminares, na tentativa de utilizar técnicas de esteganálise universal ou cega. Entretanto, devido à complexidade de se encontrar e utilizar uma ferramenta de esteganálise universal ou cega que seja eficientemente aplicada, em imagens RGB com dimensões 64×64, estes experimentos se tornaram inviáveis.

6.3 Seleção das Métricas de Qualidade

Ao final do processamento, as imagens contendo as mensagens secretas são comparadas às imagens originais, com o objetivo de validar a qualidade da imagem resultante do processo de esteganografia. Nesta comparação foram utilizadas diversas medidas de semelhança entre imagens, como: *i)* MSE, *ii)* PSNR, *iii)* SSIM, e *iv)* métricas perceptuais.

Neste trabalho as métricas MSE, PSNR e SSIM foram implementadas utilizando as bibliotecas *NumPy* e *SkImage*, ambas para linguagem *Python*. Enquanto para as métricas perceptuais foi utilizada a biblioteca *LPIPS* de código aberto, disponível no *GitHub*[6].

É importante destacar que as métricas MSE, PSNR e SSIM foram selecionadas por serem amplamente utilizadas em trabalhos presentes na literatura sobre esteganografia digital em imagens (93, 96, 102), bem como nos trabalhos de processamento de imagens (44). As métricas perceptuais foram utilizadas visando ter uma compreensão mais próxima da percepção humana sobre a distorção gerada pelas técnicas de esteganografia nas imagens pós-processadas. Optou-se por utilizar os algoritmos de métricas perceptuais ALEX e VGG por apresentarem bons resultados, de acordo com a literatura (46).

6.4 Aquisição dos *Datasets* de Imagens

Neste trabalho foram utilizadas as bases de dados *Tiny ImageNet*[7] (131) e *Pokemon Mugshots*[8]. Ambas as bases foram extraídas da plataforma *Kaggle* que é uma comunidade *online* de cientistas de dados e profissionais de aprendizado de máquina onde são compartilhados *datasets*, códigos, discussões relacionadas a área, além de hospedar competições e cursos sobre o tema.

O *dataset Tiny ImageNet* é composto por 10.000 imagens, totalizando 240 *megabytes* de fotografias diversas com dimensão 64×64. Este *dataset* foi escolhido visando avaliar a aplicação das técnicas de esteganografia apresentadas neste trabalho nas mais diversas fotografias para as quais a técnica pode ser aplicada considerando a capacidade de armazenamento de dados e intensidade da distorção gerada na imagem de capa.

Embora o *dataset Tiny ImageNet* contenha uma grande variedade de fotografias, o mesmo não contém desenho digital. Fotografias e desenhos armazenados no formato matricial apresentam pro-

[6] https://github.com/richzhang/PerceptualSimilarity

[7] https://www.kaggle.com/akash2sharma/tiny-imagenet

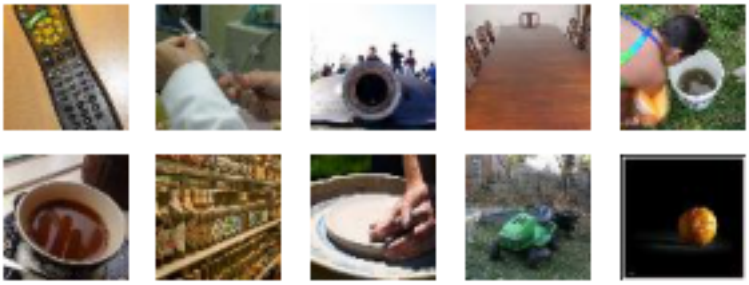
[8] https://www.kaggle.com/brilja/pokemon-mugshots-from-super-mystery-dungeon

priedades distintas, como, por exemplo, em fotografias digitais é comum que não ocorram variações abruptas de intensidade de cor entre pixels vizinhos, todavia em desenhos digitais ocorre variações abruptas em regiões específicas da imagem. Outra característica apresentada em desenhos digitais consiste em regiões da imagem compostas por pixels com a mesma cor, em fotografias digitais isto não ocorre. Estas diferenças nas categorias entre estas classes de imagens impactam o comportamento e eficiência dos algoritmos de compressão, bem como aplicação de métodos de esteganografia digital. Como técnicas de esteganografia podem ser aplicadas nas mais variadas imagens, é importante validar como a técnica proposta se comportaria quando aplicada também em ilustrações digitais. Para validar a técnica em ilustrações digitais, foi utilizado o *dataset Pokemon Mugshots*, o qual contém 4.882 imagens, totalizando 42 *megabytes* de desenhos sobre os personagens extraídos do jogo *Pokémon Super Mystery Dungeon* para o videogame *Nitendo 3DS*. Todas as imagens deste *dataset* também apresentam a dimensão 64×64.

Neste trabalho, optou-se por lidar apenas com *datasets* contendo imagens com dimensão 64×64, pois alguns dos testes realizados consomem demasiado tempo e recursos computacionais, inviabilizando o uso de dimensões maiores. Além disso, utilizamos apenas um subconjunto das imagens do *dataset*, para reduzir o tempo de processamento. Este subconjunto de imagens foi escolhido aleatoriamente para preservar as características do *dataset* inicial. Considerando a escolha de imagens com dimensões menores, inevitavelmente o espaço disponível para armazenamento da mensagem secreta também deve ser menor.

Na Figura 19 são apresentados alguns exemplos de imagens extraídos da base de dados *Tiny ImageNet*, enquanto a Figura 20 mostra exemplos de imagens extraídas da base de dados *Pokemon Mugshots*.

Figura 19: Imagens extraídas da base de dados *Tiny ImageNet*.

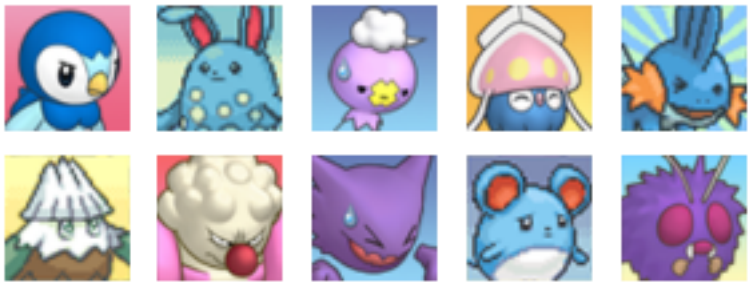


Fonte: Extraído de (132)

6.5 Implementação das Técnicas de Esteganografia

As técnicas de esteganografia selecionadas para serem comparadas neste trabalho foram implementadas utilizando-se a linguagem de programação *Python3*. Para cada categoria de técnicas foi utilizada uma abordagem diferente na implementação das mesmas, devido às suas respectivas especi-

Figura 20: Imagens extraídas da base de dados *Pokemon Mugshots*.



Fonte: Extraído de (133)

ficidades. Em todas as técnicas foi utilizada a biblioteca *SkImage* para realizar a leitura das imagens e sua conversão para representação matricial, dado que esta é uma biblioteca robusta para leitura e escrita de imagens. Além disso, foi utilizada a biblioteca *numpy* para cálculos matemáticos.

6.5.1 Esteganografia no Domínio Espacial

O conjunto de técnicas no domínio espacial é composto por: *i)* esteganografia do *bit* menos significativo (RGB e em escala de cinza), *ii)* Sistema de esteganografia do *bit* 4 e *iii)* sistema de esteganografia do *bit n*. Cada técnica deste conjunto foi aplicada a cada imagem dos dois conjuntos de imagens selecionados, sendo que em cada execução é gerada uma *string* aleatória a ser armazenada na imagem pós-processada. Ao fim do processamento, as imagens contendo esteganografia foram armazenadas utilizando-se o formato BMP, e em seguida foram calculadas as métricas de qualidade para estas imagens. É importante ressaltar que em seguida, estas imagens foram reprocessadas para realizar a extração das mensagens secretas armazenadas nas imagens. As mensagens secretas geradas de forma aleatória foram comparadas com as mensagens secretas extraídas a partir das imagens, visando avaliar o correto funcionamento das técnicas de esteganografia para armazenar e recuperar o conteúdo escondido. O algoritmo utilizado para implementar cada uma destas técnicas é apresentado no capítulo 4.

6.5.2 Esteganografia no Domínio da Frequência

A técnica de esteganografia no domínio da frequência analisada neste trabalho é baseada na transformada do cosseno discreto, sendo apresentada detalhadamente no capítulo 4. Para esta técnica é utilizado o formado JPEG para armazenar as imagens pós-processadas, dado que esta técnica foi desenvolvida explorando algumas propriedades do formato JPEG, portanto, apresenta melhores resultados quando aplicada no respectivo formato.

Dado que a primeira técnica visa armazenar os *bits* da mensagem, a ser escondida nos *bits* menos significativos dos coeficientes AC de baixa frequência de cada MCU (Unidade Codificada Miníma, do inglês, *Minimum Coded Unit*)) quantizado que compõe a imagem JPEG, e que as bibliotecas

de escrita de imagens disponíveis para linguagem *Python3* não permitem a alteração destes valores nesta etapa do algoritmo JPEG, tornou-se necessária a implementação de um codificador JPEG customizado para suportar este método de esteganografia. Além disso, torna-se também necessário a implementação de um extrator da esteganografia com base no algoritmo de um decodificador JPEG. Para este trabalho escolheu-se utilizar a versão *baseline* do algoritmo JPEG, e a versão 2.0 da especificação JFIF para implementação do codificador e extrator. Estas escolhas se devem ao fato de que estas versões são as mais utilizadas por *softwares* populares de edição de imagens, além disso, para utilização desta versão não é necessário licença de patentes, embora para algumas outras versões seja preciso. O algoritmo JPEG, bem como a especificação JFIF, foram apresentados mais detalhadamente no capítulo 2.

O codificador JPEG foi implementado utilizando a biblioteca *SkImage* para leitura das imagens e conversão para o formato matricial. Para a escrita das imagens pós-processadas, foi implementado todo o algoritmo JPEG utilizando a biblioteca *numpy* para os cálculos, e o *bitarray* resultante foi armazenado diretamente em um arquivo binário. As funções responsáveis por realizar a conversão do espaço de cores, divisão em MCUs, codificação de Huffman, DPCM, RLE e quantização foram implementadas manualmente. O extrator da esteganografia inicia seu procedimento realizando a leitura do arquivo binário JPEG para um *bitarray*, em seguida realiza um processamento inverso ao do codificador até concluir a extração da mensagem oculta na imagem.

6.5.3 Esteganografia Utilizando *Autoencoders*

Para esta técnica é utilizada a classe de *autoencoders* que é comumente utilizada para redução de ruído em imagens para esconder arquivos em imagens. Visando aproveitar o mecanismo para o qual estes autoencoders foram inicialmente desenvolvidos, o arquivo secreto é introduzido na imagem de capa através de adição de ruído. Este ruído ocupa 1/3 do tamanho da imagem de capa, sendo que cada pixel com ruído irá representar um *bit* do arquivo secreto. Optou-se por utilizar a seguinte lógica para representação dos *bits* através de ruído:

- um pixel preto representa um *bit* 0;
- um pixel branco representa um *bit* 1.

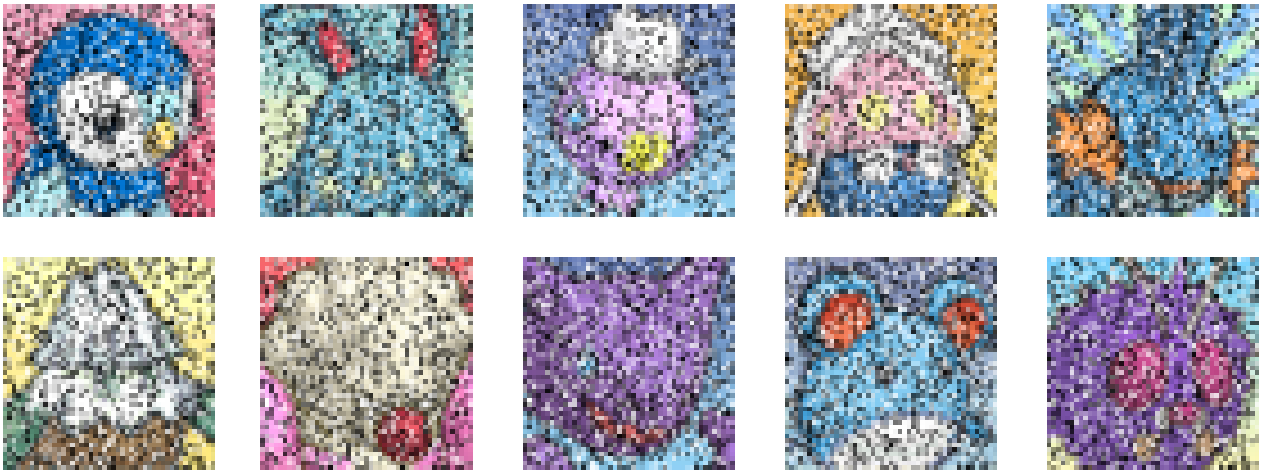
Inicialmente todas as imagens das bases de dados foram pré-processadas para adição da mensagem secreta através de ruído. As Figuras 21 e 22 apresentam alguns exemplos de imagens pré-processadas.

Para ocultar e extrair a mensagem secreta foram utilizados dois autoencoders, respectivamente. O primeiro tem o objetivo de remover o ruído da imagem, porém mantendo o significado do ruído oculto em alguma região. O segundo recupera o ruído a fim de possibilitar a extração do arquivo secreto por um pós processamento, a partir da imagem processada pelo primeiro autoencoder.

Figura 21: Exemplos de imagens pré-processadas da base de dados *Tiny ImageNet*.

Fonte: Próprio autor

Figura 22: Exemplos de imagens pré-processadas da base de dados *Pokemon Mugshots*.



Fonte: Próprio autor

Nos experimentos realizados sempre foi usada a mesma topologia para o primeiro e segundo autoencoder, sendo que foram testados autoencoders variando de uma a cinco camadas de intermediárias com conjuntos de uma a três camadas concatenadas. Os conjuntos de camadas concatenadas são compostos por camadas *Conv2d* que recebem o mesmo *input* e tem suas saídas concatenadas.

Nas Figuras 23a, 23b e 23c são apresentadas algumas das topologias utilizadas nos experimentos. Pode-se observar que a Figura 23a apresenta uma topologia composta por cinco camadas Conv2D intermediárias sem concatenação entre elas. Já na Figura 23b a topologia usada é composta por três conjuntos de camadas Conv2D intermediárias sendo concatenadas duas camadas em cada conjunto. Na Figura 23c é possível notar uma topologia composta por dois conjuntos de camadas intermediá-

rias sendo concatenadas três camadas em cada conjunto. No total foram avaliadas nove topologias distintas nos experimentos.

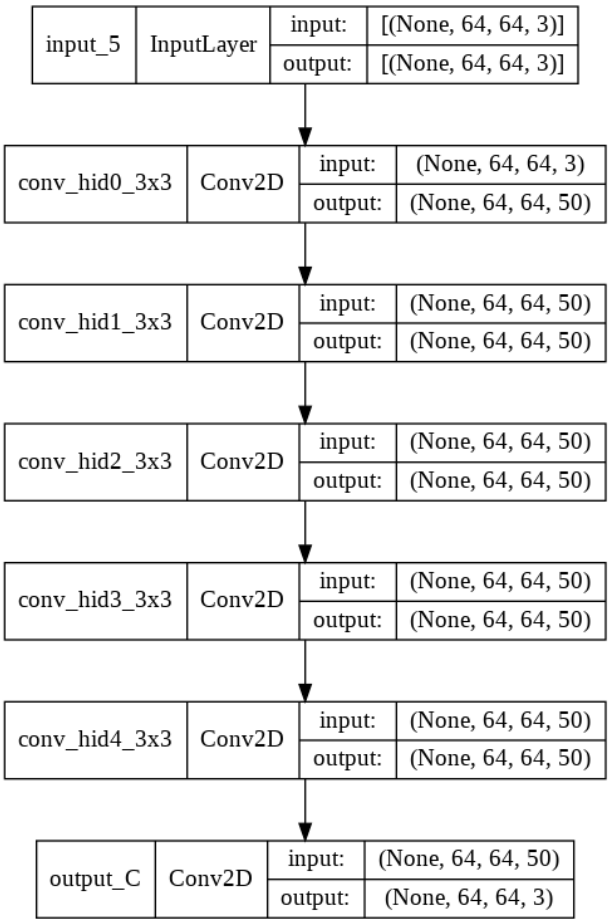
Foram realizados testes visando escolher a melhor topologia para ser usada na CNN, e em seguida foram executados outros experimentos utilizando critérios de *early-stoping*, o otimizador Adam, e definindo um número máximo de épocas a serem executadas no experimento. O valor máximo de épocas foi definido por *dataset*. Sendo igual a 100 para o *dataset Pokemon* e igual a 500 para o *dataset Tiny*. Estes experimentos foram realizados nas duas bases de dados separadamente.

Como temos dois autoencoders, para facilitar nomeamos de autoencoder1 e autoencoder2. O autoencoder1 é treinado duas vezes. No primeiro treinamento o autoencoder1 é treinado sozinho, enquanto no segundo treinamento os dois autoencoders são treinados de forma adversária. Isto é feito visando aumentar a capacidade de o autoencoder1 restaurar a imagem original.

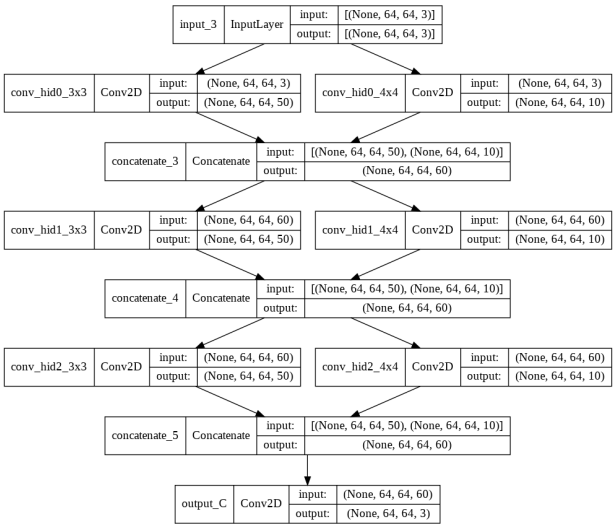
Os critérios de *early stopping* utilizados diferiram em cada um dos treinamentos. No primeiro treinamento o critério utilizado interrompe o treinamento após 10 épocas consecutivas sem melhora no valor da função de perda no conjunto de validação. Enquanto o segundo treinamento é interrompido após 5 épocas consecutivas sem melhora no valor da função de perda no conjunto de validação.

Dada a natureza estocástica do treinamento do autoencoder, em alguns casos podem surgir erros na mensagem secreta extraída da imagem pós-processada. Uma vez que que raramente a mensagem secreta é recuperada com mais de um *bit* errado, optou-se por utilizar o código de *Hamming* para identificar e corrigir o *bit* incorreto nas mensagens reveladas com até um erro. Caso a mensagem seja revelada com mais de um *bit* errado, a mesma é classificada como “mensagem corrompida” pelo sistema de esteganografia, sendo então descartada. Os procedimentos para ocultar e revelar a mensagem secreta estão descritos resumidamente nos fluxogramas das Figuras 24 e 25 respectivamente.

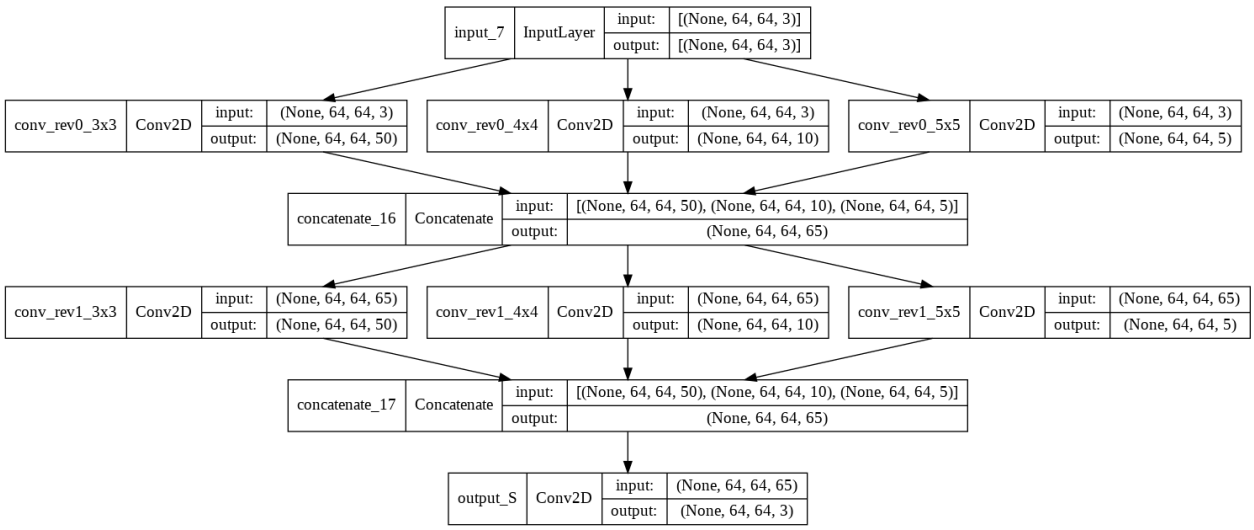
Figura 23: Alguns exemplos de topologias avaliadas.



(a) Uma camada concatenadas.



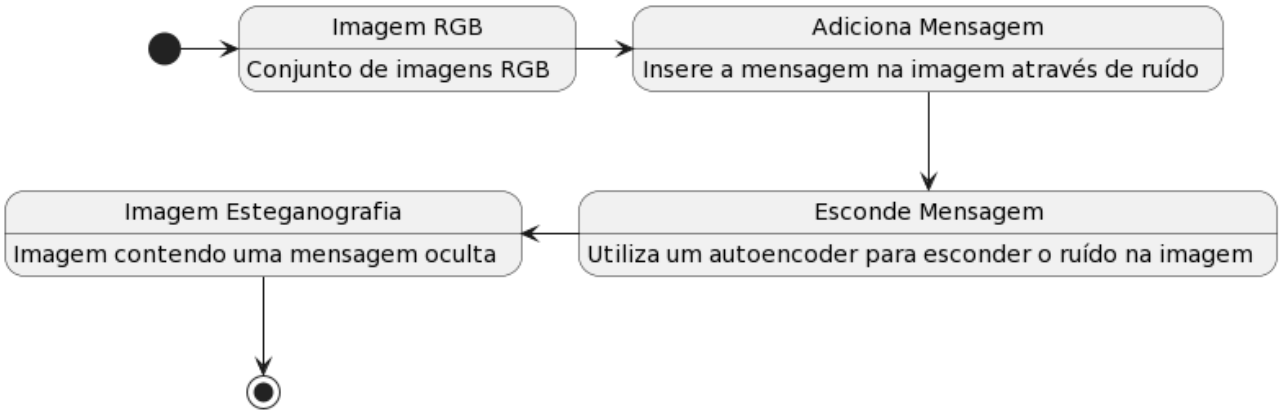
(b) Duas camadas concatenadas.



(c) Três camadas concatenadas.

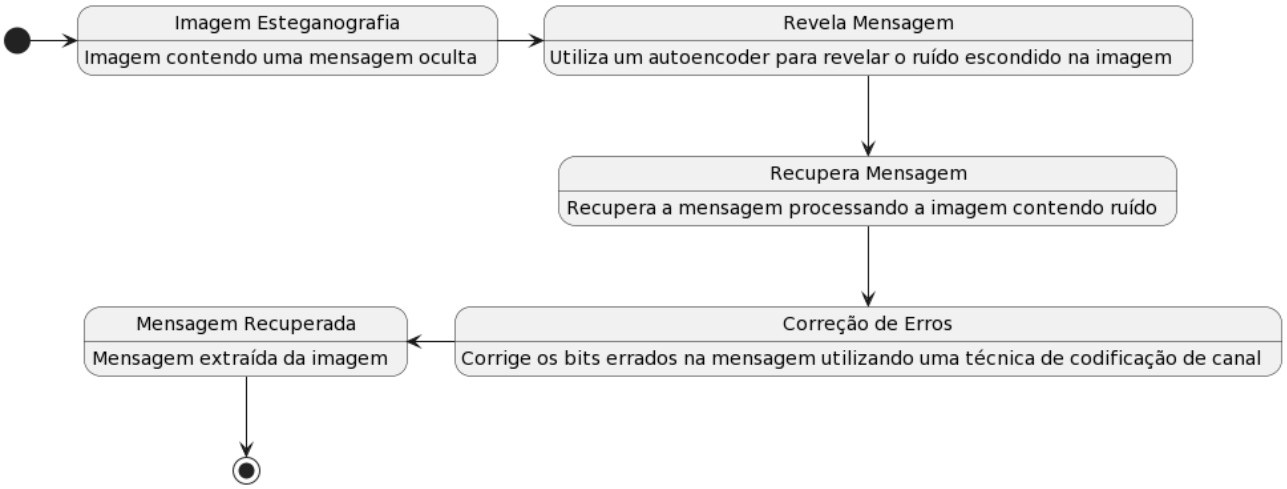
Fonte: Próprio autor

Figura 24: Procedimento para ocultar mensagem.



Fonte: Próprio autor

Figura 25: Procedimento para revelar mensagem.



Fonte: Próprio autor

7 Resultados

Neste capítulo serão apresentados e discutidos os resultados obtidos com os experimentos realizados. Os resultados são retratados por gráficos e tabelas separados por cada categoria de avaliação realizada em cada sistema de esteganografia proposto. São apresentados os resultados obtidos a partir de uma análise comparativa entre as imagens originais e as imagens pós-processadas. Nesta análise foram utilizadas medidas de semelhanças entre imagens, além de analisar também o valor absoluto da subtração entre a imagem original e a imagem pós-processada para identificar as regiões onde a mensagem secreta é armazenada no arquivo pós-processado. De modo a permitir uma comparação justa com outros sistemas de esteganografia digital, foi realizada uma análise da capacidade de armazenamento de cada sistema de esteganografia apresentado. Visando metrificar a robustez da segurança do sistema de esteganografia proposto foi efetuada uma análise de sua segurança utilizando técnicas de esteganálise do estado da arte, comparando os resultados com cada sistema de esteganografia descrito neste trabalho.

Os experimentos foram realizados no ambiente computacional da plataforma *Google Colab*. Esta plataforma fornece máquinas virtuais com capacidade para processar trabalhos em lote e algoritmos de aprendizado de máquina por um período limitado de tempo. Neste trabalho foram utilizados duas categorias de máquinas virtuais (VMs), sendo que para as técnicas de esteganografia no domínio espacial, domínio da frequência e calculo das métricas MSE, PSNR e SSIM foram utilizadas VMs com processador Intel(R) Xeon(R) CPU @ 2.20GHz, disponíveis para VM dois cores do processador, com 12 GB de memória RAM. Já para as técnicas utilizando autoencoders e calculo das métricas perceptuais é necessário utilizar uma GPU para acelerar o processamento, portanto nestes casos foram utilizadas VMs com processador Intel(R) Xeon(R) CPU @ 2.20GHz, disponíveis para VM quatro cores do processador, com 25 GB de memória RAM e GPU Nvidia Tesla P100 PCIe com 16 GB de memória RAM dedicada. O código implementado bem como as saídas retornadas durante a execução do programa foram registradas no formato de Jupyter Notebooks e encontram-se disponibilizadas publicamente no *GitHub*[9]. Utilizando-se o código implementado foi gerado um *dataset* de imagens contendo mensagens armazenadas através das teçnicas de esteganografia digital, este *dataset* encontra-se disponibilizado publicamente na plataforma *Kaggle*[10]. Em seguida são apresentados os resultados obtidos e algumas considerações sobre os mesmos.

7.1 Escolha da Topologia do Autoencoder e Técnica de Identificação e Correção de Erros

Para a escolha da melhor topologia a ser utilizada, foram executados diversos experimentos considerando diferentes topologias para cada *dataset*. A topologia com cinco conjuntos de camadas

[9] https://github.com/dzanchett/digital-steganography
[10] https://www.kaggle.com/datasets/diegozanchett/digital-steganography

intermediárias sendo concatenadas duas em cada conjunto foi escolhida para dar sequência aos experimentos com o *dataset Pokemon*. Da mesma forma, a topologia com três camadas intermediárias sendo concatenadas três em cada conjunto para dar sequência com o *dataset Tiny*. As demais topologias avaliadas estão descritas no Capítulo 6. Tais topologias foram selecionadas pois as mesmas apresentaram melhor resultado nos experimentos realizados com a respectiva base de dados.

Visando escolher a técnica de codificação de canal a ser utilizada para tratamento de erros, foi conduzida uma análise com o objetivo identificar a quantidade de *bits* errados encontrados nos dados revelados utilizando-se autoencoder. Na maioria dos experimentos realizados tal categoria de erro não foi detectada. Porém, em algumas situações a mensagem foi recuperada com um *bit* errado e na menor parte dos casos surgiram erros em mais de um *bit* da mensagem secreta pós-processada. Portanto se optou por utilizar o código de *Hamming* para tratamento de erros.

7.2 Análise das Imagens

A seguir são apresentadas as principais análises realizadas comparando a imagem original com a imagem pós-processada.

7.2.1 Medidas de Semelhanças entre Imagens

As medidas de semelhança entre imagens visam mensurar a distorção gerada na imagem pós-processada pelo sistema de esteganografia. Estas distorções foram mensuradas através de algumas funções matemáticas como o MSE, PSNR, SSIM, e também por algoritmos mais sofisticados que visam metrificar a distorção na imagem pós-processada a partir da simulação da percepção humana sobre a imagem, como os algoritmos de métricas perceptuais ALEX e VGG. Na Tabela 2 são apresentados os resultados médios obtidos utilizando-se as métricas MSE, PSNR e SSIM para comparar as imagens originais e pós-processadas. Nesta tabela também são apresentados os resultados numéricos para as métricas perceptuais obtidas a partir das redes neurais ALEX e VGG.

Tabela 2: Medidas de Semelhanças entre Imagens.

	Métrica									
	MSE		**PSNR**		**SSIM**		**ALEX**		**VGG**	
Técnica	**Pokemon**	**Tiny**	**Pokemon**	**Tiny**	**Pokemon**	**Tiny**	**Pokemon**	**Tiny**	**Pokemon**	**Tiny**
LSB	2.4363	2.3405	44.2707	44.4391	**1.0**	**1.0**	**0.0001**	**0.0002**	**0.0060**	**0.0046**
LSB Escala de Cinza	**2.3029**	**2.3049**	**44.5111**	**44.5057**	**1.0**	**1.0**	-	-	-	-
SSB-4	13.7901	13.3729	36.7615	36.8829	**1.0**	**1.0**	0.0009	0.0018	0.0311	0.0248
SSB-N	4.8874	4.7683	41.2560	41.3581	**1.0**	**1.0**	0.0002	0.0005	0.0175	0.0108
DCT	139.5231	61.8187	26.4855	31.4538	0.9836	0.9975	0.0058	0.0034	0.0957	0.0271
Autoencoder	146.2502	271.2275	26.6755	24.5713	**1.0**	**1.0**	0.0213	0.0442	0.1409	0.1356

De acordo com os dados apresentados na Tabela 2, observa-se que a técnica que apresentou os melhores valores para métrica MSE e PSNR foi a LSB em escala de cinza, seguida da técnica LSB. Isto ocorre pelo fato de ambas as técnicas realizarem alterações apenas nos dois *bits* menos significativos de cada *byte*, gerando portanto um impacto limitado na imagem pós-processada. Para

técnica SSIM, dado o comportamento da fórmula que visa estimar a degradação estrutural da imagem, quase todas as técnicas testadas apresentaram um bom resultado, sendo DCT a única técnica com um resultado inferior. Para métricas perceptuais a técnica LSB segue com o melhor resultando, sendo que em seguida vem a técnica SSB-N. A técnica LSB em Escala de Cinza não foi avaliada utilizando-se métricas perceptuais devido a uma limitação da biblioteca utilizada neste trabalho.

7.2.2 Diferença entre Imagens — Locais onde a Mensagem Secreta é Armazenada.

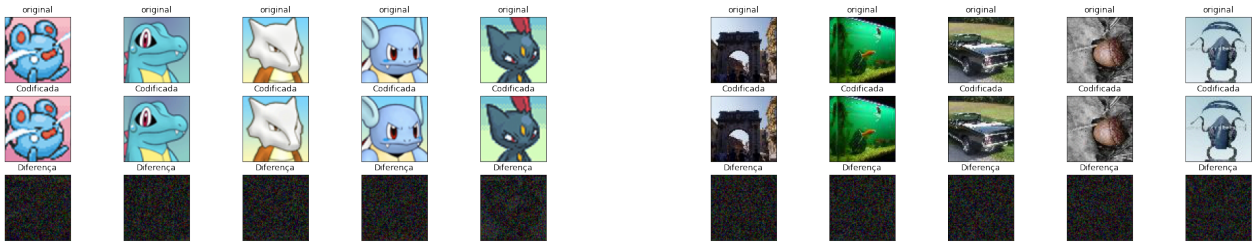
Visando identificar as regiões onde a mensagem secreta encontra-se armazenada na imagem pós-processada foi utilizada uma técnica que consiste em plotar o valor absoluto da diferença entre a imagem original e a pós-processada. Com isto é gerada uma nova imagem onde os pixels que tiverem valores mais distantes na comparação entre ambas imagens são representados em cores mais claras, enquanto os pixels que tiverem valores mais próximos são representados em cores mais escuras. É importante ressaltar que os pixels que contêm valores mais distantes nesta análise são os pixels que carregam mais informações do arquivo secreto.

Foram selecionadas 5 imagens de cada *dataset* para realizar a análise da diferença, sendo executada para cada técnica em comparação. Para as Figuras 26, 27, 28, 29, 30 e 31 apresentam na primeira linha a imagem original extraída do respectivo *dataset*, na segunda linha a imagem modificada para conter os dados ocultos utilizando a respectiva técnica de esteganografia e na terceira linha a imagem resultante da análise da diferença entre às imagens das duas primeiras linhas. Para LSB (Figura 26), LSB Escala de Cinza (Figura 27) e SSB-4 (Figura 28) nota-se que a mensagem secreta encontra-se armazenada com a mesma intensidade por toda a imagem de capa, ou seja, é possível visualizar um ruído de energia aproximadamente constante ao observar a imagem que representa o *plot* do valor absoluto da diferença entre a imagem original e pós-processada. Já para técnica SSB-N (Figura 29) nota-se que este ruído varia de intensidade aleatoriamente ao longo da imagem. Para técnica baseada em DCT (Figura 30) nota-se que a diferença entre as imagens é mais acentuada em desenhos (*dataset Pokemon Mugshots*) do que em fotografia (*dataset Tiny Imagenet*). Isso se deve ao fato do algoritmo do formato JPEG, o qual é utilizado no armazenamento de imagens com esta técnica de esteganografia, apresentar melhores resultados em fotografias quando comparado a desenhos. Já a técnica baseada em autoencoders (Figura 31) armazena o conteúdo oculto nas quinas da imagem, ou seja, nas linhas do desenho ou nas regiões de mudança de textura nas fotografias.

7.3 Análise da Capacidade de Armazenamento do Sistema de Esteganografia

Um dos principais objetivos da utilização da esteganografia é ocultar a transmissão ou o armazenamento de um dado. Portanto, para evitar o consumo desnecessário de recursos computacionais, é importante que as técnicas de esteganografia evoluam de modo a permitir armazenar uma quantidade crescente de informação dentro de um arquivo redundante. Todavia, é necessário também garantir a segurança do dado ocultado e reduzir a distorção gerada no arquivo redundante. Visando identificar

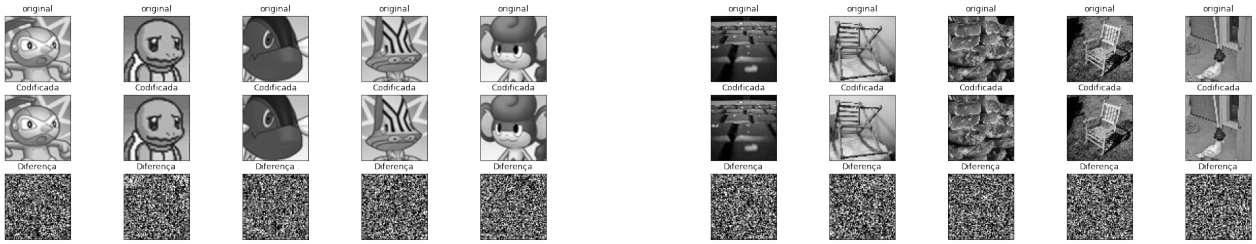
Figura 26: Diferença Entre Imagens - LSB.



(a) *Dataset Pokemon Mugshots*. (b) *Dataset Tiny ImageNet*.

Fonte: Próprio autor

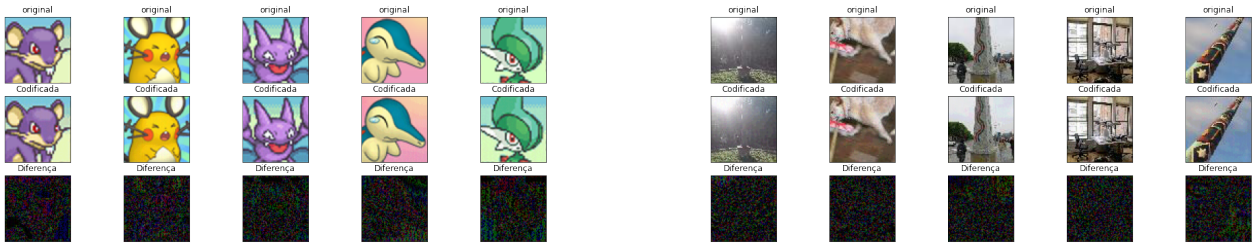
Figura 27: Diferença Entre Imagens - LSB Escala de Cinza.



(a) *Dataset Pokemon Mugshots*. (b) *Dataset Tiny ImageNet*.

Fonte: Próprio autor

Figura 28: Diferença Entre Imagens - SSB-4.



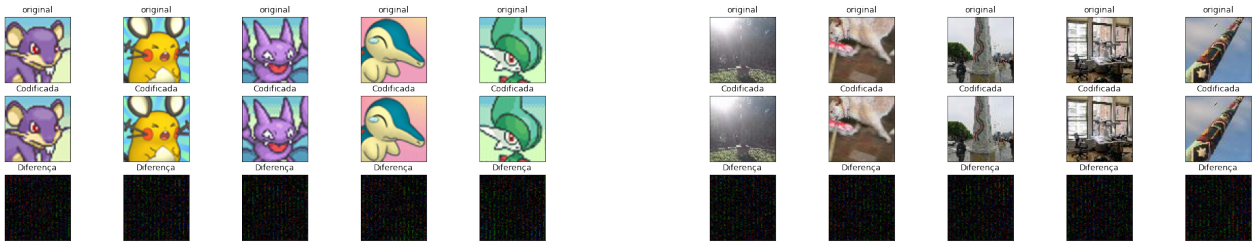
(a) *Dataset Pokemon Mugshots*. (b) *Dataset Tiny ImageNet*.

Fonte: Próprio autor

esta característica nas técnicas, foi realizada uma análise da capacidade de armazenamento de cada técnica apresentada.

Para cada técnica de esteganografia apresentada neste trabalho foi realizada uma observação visando definir a sua respectiva capacidade máxima de armazenamento. Os resultados obtidos encontram-se apresentados na Tabela 3. Observa-se que as técnicas que apresentam maior capacidade de armazenamento são as técnicas LSB e LSB em escala de cinza, permitindo armazenar 2 *bits* por *byte*. Em seguida vem as técnicas SSB-4 e SSB-N que permitem armazenar 1 *bit* por *byte*. Já a técnica baseada em autoencoder armazena 0.33 *bit* por *byte* e a técnica baseada em DCT 0.015 *bit* por *byte*.

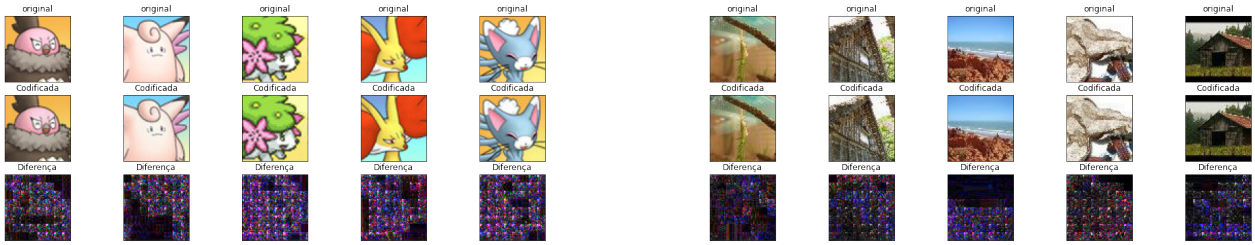
Figura 29: Diferença Entre Imagens - SSB-N.



(a) *Dataset Pokemon Mugshots*. (b) *Dataset Tiny ImageNet*.

Fonte: Próprio autor

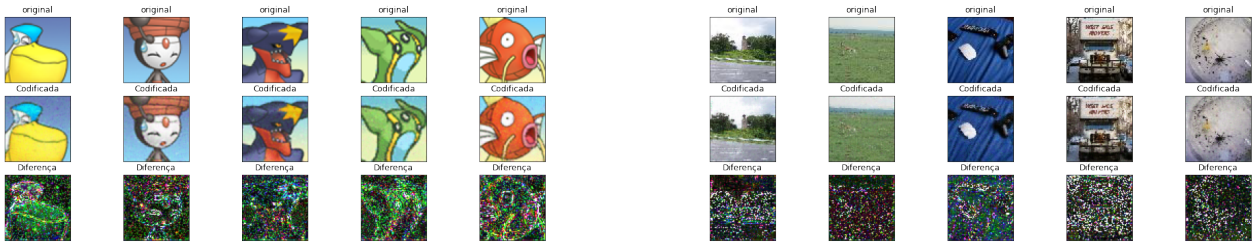
Figura 30: Diferença Entre Imagens - DCT.



(a) *Dataset Pokemon Mugshots*. (b) *Dataset Tiny ImageNet*.

Fonte: Próprio autor

Figura 31: Diferença Entre Imagens - Autoencoder.



(a) *Dataset Pokemon Mugshots*. (b) *Dataset Tiny ImageNet*.

Fonte: Próprio autor

Tabela 3: Capacidade de Armazenamento do Sistema de Esteganografia.

Técnica	**Capacidade de Armazenamento**
LSB	2 *bits* por *byte*
LSB Escala de Cinza	2 *bits* por *byte*
SSB-4	1 *bit* por *byte*
SSB-N	1 *bit* por *byte*
Autoencoder	0.33 *bit* por *byte*
DCT	0.015 *bit* por *byte*

7.4 Análise da Segurança do Sistema de Esteganografia

Visando validar a segurança do sistema de esteganografia proposto foram realizados uma série de testes utilizando uma ferramenta de esteganálise. Avaliou-se a acurácia desta ferramenta ao classificar se imagens pós-processadas continham ou não dados armazenados por cada procedimento de esteganografia em análise neste trabalho.

Foram executados um experimento em cada conjunto de imagens pós-processadas pelas técnicas de esteganografia apresentadas neste trabalho, que consiste em fornecer um conjunto de imagens contendo esteganografia para a ferramenta de esteganálise. O objetivo é verificar o percentual de imagens classificadas como suspeitas, que são imagens onde pode haver esteganografia.

Para técnica LSB foram classificadas $4,71\%$ das imagens do *dataset Pokemon Mugshots* como suspeitas, enquanto para o *dataset Tiny Imagenet* $50,01\%$ das imagens são suspeitas. Podemos concluir que nos experimentos realizados esta técnica de esteganografia funcionou melhor com desenhos em comparação a fotografias. Já para técnica LSB em escala de cinza foram obtidos resultados semelhantes para os dois *datasets*, sendo que para o conjunto de imagens *Pokemon Mugshots* foram classificadas $49,30\%$ como suspeitas, enquanto para o outro conjunto $64,08\%$ das imagens foram classificadas como suspeitas.

As técnicas SSB-4 e SSB-N apresentaram os piores resultados em relação à segurança com base nesta ferramenta de esteganálise. É importante ressaltar que este é um experimento realizado com uma única ferramenta de esteganálise que emprega diversas técnicas distintas. Por esse motivo, é significativo realizar experimentos com outras ferramentas para chegar em conclusões mais precisas em relação à segurança. A técnica SSB-4 apresentou um total de 100% das imagens do *dataset Pokemon* classificadas como suspeitas, enquanto para o outro dataset $99,22\%$ das imagens foram classificadas como suspeitas. Já para a técnica SSB-N $98,19\%$ das imagens do *conjunto Pokemon* foram classificadas como suspeitas, enquanto para o outro conjunto $98,25\%$ das imagens.

A técnica baseada na transformada do cosseno discreto apresentou resultados razoáveis, sendo que para o *dataset Pokemon* $76,60\%$ das imagens foram classificadas como suspeitas, enquanto para o outro *dataset* $69,98\%$ das imagens são suspeitas. Já a técnica baseada em autoencoder apresentou um bom resultado, assim como a técnica LSB. Para técnica baseada em autoencoder no *dataset Pokemon* $22,93\%$ das imagens são classificadas como suspeitas, e $53,8\%$ das imagens do *dataset Tiny* são suspeitas.

Um experimento reverso foi realizado, o qual consiste em fornecer um conjunto de imagens sem esteganografia para a ferramenta de esteganálise. Assim como nos experimentos anteriores, avaliou-se o percentual de imagens classificadas como suspeitas, logo para este, quanto menor o valor melhor o desempenho da ferramenta. No *dataset Pokemon Mugshots* o percentual de imagens suspeitas encontrado foi de $4,5\%$, enquanto no *dataset Tiny ImageNet* o percentual encontrado foi de $56,9\%$.

Com isto observa-se que a ferramenta de esteganálise apresentou no sistema de esteganografia

proposto uma acurácia pouco significativa, próxima ou abaixo a 50% que representa o resultado obtido por um classificador aleatório, demonstrando assim a segurança do sistema de esteganografia proposto em relação a esta ferramenta de esteganálise.

Tabela 4: Segurança do Sistema de Esteganografia.

	Percentual de Imagens Detectadas	
Técnica	**Pokemon**	**Tiny**
LSB	**4,71**	**50,01**
LSB Escala de Cinza	49,30	64,08
SSB-4	100,0	99,22
SSB-N	98,19	98,25
DCT	76,60	69,98
Autoencoder	22,93	53,8

7.5 Análise do Tempo de Processamento de cada *Dataset* para cada Técnica

Também foi armazenado o tempo em segundos, necessário para processar todo o *dataset* com cada técnica de esteganografia, além do tempo médio de processamento para cada imagem. Estes resultados são apresentados na Tabela 5.

A técnica que apresentou o menor de tempo de processamento foi LSB Escala de Cinza. Este fato está relacionado à menor quantidade de canais em imagens em escala de cinza. Sendo assim, a quantidade de informações que o algoritmo irá processar é reduzida. Em seguida vêm as técnicas SSB-4, SSB-N e DCT, seguidas pela técnica LSB. A técnica que consumiu mais tempo para processar foi a técnica baseada em Autoencoder. Isto se deve ao fato da mesma ser implementada com base em uma rede neural, que apresenta um alto consumo de recursos computacionais, e necessita de tempo para realizar seu treinamento.

Tabela 5: Tempo de Processamento do Sistema de Esteganografia.

	Tempo de Processamento Dataset (Segundos)		**Tempo de Processamento por Imagem (Segundos)**	
Técnica	**Pokemon**	**Tiny**	**Pokemon**	**Tiny**
LSB	1483,28	2130,73	0,30	0,21
LSB Escala de Cinza	**566,51**	**558,03**	**0,11**	**0,11**
SSB-4	984,44	1057,58	0,20	0,21
SSB-N	1011,35	1039,86	0,20	0,21
DCT	1091,16	966,66	0,22	0,19
Autoencoder	1493,77	11187,62	1,22	11,18

8 Conclusão

O objetivo deste trabalho consiste em avaliar diversas técnicas para aplicação de esteganografia em imagem. Algumas destas técnicas foram extraídas da literatura. Todavia a técnica de esteganografia baseada em *autoencoders* foi proposta pelo autor do trabalho.

Todas as técnicas apresentadas neste trabalho foram implementadas utilizando-se a linguagem de programação *Python*, e utilizadas para armazenar *strings* geradas aleatoriamente em imagens de dois *datasets* extraídos da plataforma *Kaggle*. Após o processamento das imagens foram realizados análises utilizando métricas adequadas para avaliar a qualidade das imagens pós-processadas, bem como a segurança e eficiencia das técnicas de esteganografia apresentadas.

Foram comparadas diversas técnicas de esteganografia digital presentes na literatura, além da técnica proposta, a qual apresenta simultaneamente características de técnicas de esteganografia com incorporação e sem incorporação. Para isto, foi realizada uma revisão bibliográfica visando selecionar técnicas para se aplicar neste trabalho. Foram eleitas 6 técnicas de esteganografia, as quais foram implementadas utilizando a linguagem de programação Python. Cinco técnicas foram extraídas da literatura, enquanto uma foi proposta pelo autor. Os experimentos computacionais foram realizados através da aplicação nas imagens de dois *datasets* que com características distintas. Além disto foram realizadas análises: *i)* na qualidade das imagens pós-processadas, *ii)* na segurança do sistema de esteganografia em relação a sistemas de esteganálise, *iii)* tempo de processamento necessário e *iv)* capacidade de armazenamento dos sistemas de esteganografia.

Os resultados obtidos apresentam diferenças em relação às propriedades avaliadas. Em relação à qualidade da imagem pós-processada a técnica sistema de esteganografia LSB em escala de cinza apresentou o melhor resultado. Entretanto, considerando a capacidade de armazenamento do sistema de esteganografia, a melhor técnica se baseia na utilização dos dois *bits* menos significativos para armazenar a mensagem oculta, ou seja, LSB ou LSB em escala de cinza. Já com relação ao tempo de processamento a técnica vencedora consiste no sistema de esteganografia LSB escala de cinza. Considerando a segurança do sistema de esteganografia a técnica LSB superou as demais, e em seguida o método baseado em autoencoder apresentou melhor resultado.

Em trabalhos futuros pode ser interessante a comparação de mais técnicas de esteganografia em diferentes domínios. Como exemplo, a esteganografia baseada na transformada *wavelet*. Dado que algumas técnicas de esteganografia mais recentes apresentam certa resistência a algumas categorias de edições nas imagens, esta propriedade poderia ser avaliada através da aplicação de filtros e ruídos nas imagens pós-processadas antes da extração da mensagem oculta. Além disso, visando aprimorar a técnica de esteganografia baseada em autoencoders, poderiam ser avaliados outros algoritmos de remoção de ruído de imagens.

REFERÊNCIAS

1 SHARMA, A. Trends in internet-based business-to-business marketing. **Industrial marketing management**, Elsevier, v. 31, n. 2, p. 77–84, 2002.

2 ODLYZKO, A. Privacy, economics, and price discrimination on the internet. In: **Economics of information security**. [S.l.]: Springer, 2004. p. 187–211.

3 CENTER, I. A. **Global Data Leakage Report, 2013**. 2015.

4 ZHANG, X.; GHORBANI, A. A. An overview of online fake news: Characterization, detection, and discussion. **Information Processing & Management**, Elsevier, v. 57, n. 2, p. 102025, 2020.

5 AGARWAL, S. et al. Protecting world leaders against deep fakes. In: **CVPR workshops**. [S.l.: s.n.], 2019. v. 1.

6 JENIK, C. **Infographic: A Minute on the Internet in 2021**. 2021. Disponível em: https://www.statista.com/chart/25443/estimated-amount-of-data-created-on-the-internet-in-one-minute/.

7 YASSEIN, M. B. et al. Comprehensive study of symmetric key and asymmetric key encryption algorithms. In: IEEE. **2017 international conference on engineering and technology (ICET)**. [S.l.], 2017. p. 1–7.

8 SARMAH, D. K.; BAJPAI, N. Proposed system for data hiding using cryptography and steganography. **International Journal of Computer Applications**, International Journal of Computer Applications, 244 5 th Avenue,# 1526, New . . . , v. 8, n. 9, p. 7–10, 2010.

9 SIPER, A.; FARLEY, R.; LOMBARDO, C. The rise of steganography. **Proceedings of student/faculty research day, CSIS, Pace University**, 2005.

10 DOSHI, R.; JAIN, P.; GUPTA, L. Steganography and its applications in security. **International Journal of Modern Engineering Research (IJMER)**, v. 2, n. 6, p. 4634–4638, 2012.

11 JARUSEK, R.; VOLNA, E.; KOTYRBA, M. Photomontage detection using steganography technique based on a neural network. **Neural Networks**, Elsevier, v. 116, p. 150–165, 2019.

12 KHAIRINA, N.; HARAHAP, M. K.; LUBIS, J. H. The authenticity of image using hash md5 and steganography least significant bit. **IJISTECH (International Journal of Information System & Technology)**, v. 2, n. 1, p. 1–6, 2018.

13 MORKEL, T.; ELOFF, J. H.; OLIVIER, M. S. An overview of image steganography. In: **ISSA**. [S.l.: s.n.], 2005. v. 1, n. 2, p. 1–11.

14 O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. **arXiv preprint arXiv:1511.08458**, 2015.

15 GU, J. et al. Recent advances in convolutional neural networks. **Pattern Recognition**, Elsevier, v. 77, p. 354–377, 2018.

16 O'SULLIVAN, A. C.; THIERER, A. D. Projecting the growth and economic impact of the internet of things. **Available at SSRN 2618794**, 2015.

17 DEY, S.; ROY, A.; DAS, S. Home automation using internet of thing. In: IEEE. **2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)**. [S.l.], 2016. p. 1–6.

18 ANDERSON, R. J.; PETITCOLAS, F. A. On the limits of steganography. **IEEE Journal on selected areas in communications**, IEEE, v. 16, n. 4, p. 474–481, 1998.

19 PROVOS, N.; HONEYMAN, P. Hide and seek: An introduction to steganography. **IEEE security & privacy**, IEEE, v. 1, n. 3, p. 32–44, 2003.

20 HU, D. et al. A novel image steganography method via deep convolutional generative adversarial networks. **IEEE Access**, IEEE, v. 6, p. 38303–38314, 2018.

21 WU, S.; ZHONG, S.; LIU, Y. Deep residual learning for image steganalysis. **Multimedia tools and applications**, Springer, v. 77, n. 9, p. 10437–10453, 2018.

22 REDDY, P. et al. Im2vec: Synthesizing vector graphics without vector supervision. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2021. p. 7342–7351.

23 WATT, A.; LILLEY, C. **SVG unleashed**. [S.l.]: Sams Publishing, 2002.

24 JOSHI, M. A. **Digital image processing: An algorithmic approach**. [S.l.]: PHI Learning Pvt. Ltd., 2018.

25 BHABATOSH, C. et al. **Digital image processing and analysis**. [S.l.]: PHI Learning Pvt. Ltd., 1977.

26 CHEDDAD, A. et al. Digital image steganography: Survey and analysis of current methods. **Signal processing**, Elsevier, v. 90, n. 3, p. 727–752, 2010.

27 FORD, A.; ROBERTS, A. Colour space conversions. **Westminster University, London**, v. 1998, p. 1–31, 1998.

28 IBRAHEEM, N. A. et al. Understanding color models: a review. **ARPN Journal of science and technology**, Citeseer, v. 2, n. 3, p. 265–275, 2012.

29 WELLS, D. C.; GREISEN, E. W. Fits-a flexible image transport system. In: **Image Processing in Astronomy**. [S.l.: s.n.], 1979. p. 445.

30 MUSTRA, M.; DELAC, K.; GRGIC, M. Overview of the dicom standard. In: IEEE. **2008 50th International Symposium ELMAR**. [S.l.], 2008. v. 1, p. 39–44.

31 FERRAIOLO, J.; JUN, F.; JACKSON, D. **Scalable vector graphics (SVG) 1.0 specification**. [S.l.]: iuniverse Bloomington, 2000.

32 MIANO, J. **Compressed image file formats: Jpeg, png, gif, xbm, bmp**. [S.l.]: Addison-Wesley Professional, 1999.

33 SITIO, A. S. Text message compression analysis using the lz77 algorithm. **INFOKUM**, v. 7, n. 1, Desembe, p. 16–21, 2018.

34 ROELOFS, G. **PNG: the definitive guide**. [S.l.]: O'Reilly Media, 1999.

35 ISO, I. Iec 10918-1. itu-t recommendation t. 81. **Information Technology-Digital Compression and Coding of Continuous-tone Still Image-Requirements and Guidelines**, 1993.

36 HAMILTON, E. Jpeg file interchange format. 2004.

37 KHALID, M. Y. U. 2020. Disponível em: https://yasoob.me/posts/understanding-and-writing-jpeg-decoder-in-python/.

38 KATZ, J.; LINDELL, Y. **Introduction to modern cryptography**. [S.l.]: CRC press, 2020.

39 BERLEKAMP, E. R. The technology of error-correcting codes. **Proceedings of the IEEE**, IEEE, v. 68, n. 5, p. 564–593, 1980.

40 SINGH, A. K. Error detection and correction by hamming code. In: IEEE. **2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)**. [S.l.], 2016. p. 35–37.

41 SKLAR, B. Reed-solomon codes. **Downloaded from URL http://www. informit. com/content/images/art. sub.–sklar7. sub.–reed-solomo-n/elementLinks/art. sub.–sklar7. sub.–reed-solomon. pdf**, p. 1–33, 2001.

42 NGUYEN, J. P. **Applications of Reed-Solomon codes on optical media storage**. Tese (Doutorado) — Citeseer, 2011.

43 MCELIECE, R. J.; SWANSON, L. Reed-solomon codes and the exploration of the solar system. 1994.

44 SARA, U.; AKTER, M.; UDDIN, M. S. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. **Journal of Computer and Communications**, Scientific Research Publishing, v. 7, n. 3, p. 8–18, 2019.

45 PRASHNANI, E. et al. Pieapp: Perceptual image-error assessment through pairwise preference. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 1808–1817.

46 ZHANG, R. et al. The unreasonable effectiveness of deep features as a perceptual metric. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 586–595.

47 LEGG, S.; HUTTER, M. et al. A collection of definitions of intelligence. **Frontiers in Artificial Intelligence and applications**, IOS press, v. 157, p. 17, 2007.

48 ANDERSON, M.; REID, C. Intelligence. **MS Encarta online encyclopedia**, 2006.

49 ALBUS, J. S. Outline for a theory of intelligence. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 21, n. 3, p. 473–509, 1991.

50 TURING, A. M. Computing machinery and intelligence. In: **Parsing the turing test**. [S.l.]: Springer, 2009. p. 23–65.

51 MCCARTHY, J. What is artificial intelligence. **URL: http://www-formal. stanford. edu/jmc/whatisai. html**, 2004.

52 SHUBHENDU, S. S.; VIJAY, J. Applicability of artificial intelligence in different fields of life. **International Journal of Scientific Engineering and Research**, v. 1, n. 1, p. 28–35, 2013.

53 FLOWERS, J. C. Strong and weak ai: Deweyan considerations. In: **AAAI Spring Symposium: Towards Conscious AI Systems**. [S.l.: s.n.], 2019. v. 22877.

54 LIU, B. " weak ai" is likely to never become" strong ai", so what is its greatest value for us? **arXiv preprint arXiv:2103.15294**, 2021.

55 NG, G. W.; LEUNG, W. C. Strong artificial intelligence and consciousness. **Journal of Artificial Intelligence and Consciousness**, World Scientific, v. 7, n. 01, p. 63–72, 2020.

56 SUN, R. Artificial intelligence: Connectionist and symbolic approaches. Citeseer, 1999.

57 MURPHY, K. P. **Machine learning: a probabilistic perspective**. [S.l.]: MIT press, 2012.

58 GREFENSTETTE, J. J. Genetic algorithms and machine learning. In: **Proceedings of the sixth annual conference on Computational learning theory**. [S.l.: s.n.], 1993. p. 3–4.

59 BISHOP, C. M. Neural networks and their applications. **Review of scientific instruments**, American Institute of Physics, v. 65, n. 6, p. 1803–1832, 1994.

60 SANTOS, V. S. d. **Neurônio: O que É, tipos, função, estrutura**. Mundo Educação. Disponível em: https://mundoeducacao.uol.com.br/biologia/neuronios.htm.

61 MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

62 HAYMAN, S. The mcculloch-pitts model. In: IEEE. **IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)**. [S.l.], 1999. v. 6, p. 4438–4439.

63 KUBAT, M. Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7. **The Knowledge Engineering Review**, Cambridge University Press, v. 13, n. 4, p. 409–412, 1999.

64 MERCIONI, M. A.; HOLBAN, S. The most used activation functions: Classic versus current. In: IEEE. **2020 International Conference on Development and Application Systems (DAS)**. [S.l.], 2020. p. 141–145.

65 RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach. 2002.

66 SHARMA, S.; SHARMA, S.; ATHAIYA, A. Activation functions in neural networks. **towards data science**, v. 6, n. 12, p. 310–316, 2017.

67 KLAMBAUER, G. et al. Self-normalizing neural networks. **Advances in neural information processing systems**, v. 30, 2017.

68 CUNNINGHAM, P.; CORD, M.; DELANY, S. J. Supervised learning. In: **Machine learning techniques for multimedia**. [S.l.]: Springer, 2008. p. 21–49.

69 DAYAN, P.; SAHANI, M.; DEBACK, G. Unsupervised learning. **The MIT encyclopedia of the cognitive sciences**, MIT Press, p. 857–859, 1999.

70 KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. **Journal of artificial intelligence research**, v. 4, p. 237–285, 1996.

71 SADEEQ, M. A.; ABDULAZEEZ, A. M. Neural networks architectures design, and applications: A review. In: IEEE. **2020 International Conference on Advanced Science and Engineering (ICOASE)**. [S.l.], 2020. p. 199–204.

72 TAUD, H.; MAS, J. Multilayer perceptron (mlp). In: **Geomatic approaches for modeling land change scenarios**. [S.l.]: Springer, 2018. p. 451–455.

73 MEDSKER, L.; JAIN, L. Recurrent neural networks. Citeseer.

74 BANK, D.; KOENIGSTEIN, N.; GIRYES, R. Autoencoders. **arXiv preprint arXiv:2003.05991**, 2020.

75 GOODFELLOW, I. et al. Generative adversarial nets. **Advances in neural information processing systems**, v. 27, 2014.

76 CRESWELL, A. et al. Generative adversarial networks: An overview. **IEEE Signal Processing Magazine**, IEEE, v. 35, n. 1, p. 53–65, 2018.

77 ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: IEEE. **2017 International Conference on Engineering and Technology (ICET)**. [S.l.], 2017. p. 1–6.

78 ZIELIŃSKA, E.; MAZURCZYK, W.; SZCZYPIORSKI, K. Trends in steganography. **Communications of the ACM**, ACM New York, NY, USA, v. 57, n. 3, p. 86–95, 2014.

79 FRIDRICH, J.; GOLJAN, M. Practical steganalysis of digital images: state of the art. **security and Watermarking of Multimedia Contents IV**, SPIE, v. 4675, p. 1–13, 2002.

80 KAUR, N.; BEHAL, S. A survey on various types of steganography and analysis of hiding techniques. **International journal of engineering trends and technology**, v. 11, n. 8, p. 388–392, 2014.

81 ZHOU, Z. et al. Coverless image steganography without embedding. In: SPRINGER. **International Conference on Cloud Computing and Security**. [S.l.], 2015. p. 123–132.

82 WU, K.-C.; WANG, C.-M. Steganography using reversible texture synthesis. **IEEE Transactions on Image Processing**, IEEE, v. 24, n. 1, p. 130–139, 2014.

83 KADHIM, I. J. et al. Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. **Neurocomputing**, Elsevier, v. 335, p. 299–326, 2019.

84 PETITCOLAS, F. A.; ANDERSON, R. J.; KUHN, M. G. Information hiding-a survey. **Proceedings of the IEEE**, IEEE, v. 87, n. 7, p. 1062–1078, 1999.

85 WAYNER, P. **Disappearing Cryptography: Information Hiding: Steganography & Watermarking**. [S.l.]: Morgan Kaufmann, 2002.

86 TIWARI, N.; SHANDILYA, D. M. Evaluation of various lsb based methods of image steganography on gif file format. **International Journal of Computer Applications**, International Journal of Computer Applications, 244 5 th Avenue,# 1526, New . . . , v. 6, n. 2, p. 1–4, 2010.

87 BT, I.-R. R. Basic parameter values for the hdtv standard for the studio and for international programme exchange. ITU Geneva, 1990.

88 RODRIGUES, J. M.; RIOS, J.; PUECH, W. Ssb-4 system of steganography using bit 4. In: **WIAMIS: Workshop on Image Analysis for Multimedia Interactive Services**. [S.l.: s.n.], 2004.

89 ZANCHETT, D. et al. Análise comparativa de métodos para esteganografia digital em imagens. **Anais do Computer on the Beach**, v. 12, p. 240–247, 2021.

90 AHMED, N.; NATARAJAN, T.; RAO, K. R. Discrete cosine transform. **IEEE transactions on Computers**, IEEE, v. 100, n. 1, p. 90–93, 1974.

91 KARAMPIDIS, K.; KAVALLIERATOU, E.; PAPADOURAKIS, G. A review of image steganalysis techniques for digital forensics. **Journal of information security and applications**, Elsevier, v. 40, p. 217–235, 2018.

92 LAMSHÖFT, K. et al. Knock, knock, log: Threat analysis, detection & mitigation of covert channels in syslog using port scans as cover. **Forensic Science International: Digital Investigation**, Elsevier, v. 40, p. 301335, 2022.

93 YOUNUS, Z. S.; HUSSAIN, M. K. Image steganography using exploiting modification direction for compressed encrypted data. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, 2019.

94 GRAJEDA, C.; BREITINGER, F.; BAGGILI, I. Availability of datasets for digital forensics–and what is missing. **Digital Investigation**, Elsevier, v. 22, p. S94–S105, 2017.

95 GITTINS, Z.; SOLTYS, M. Malware persistence mechanisms. **Procedia Computer Science**, Elsevier, v. 176, p. 88–97, 2020.

96 SHAH, P. D.; BICHKAR, R. S. Secret data modification based image steganography technique using genetic algorithm having a flexible chromosome structure. **Engineering Science and Technology, an International Journal**, Elsevier, v. 24, n. 3, p. 782–794, 2021.

97 RAJESWARI, J.; JAGANNATH, M. Advances in biomedical signal and image processing–a systematic review. **Informatics in Medicine Unlocked**, Elsevier, v. 8, p. 13–19, 2017.

98 KHEDMATI, Y.; PARVAZ, R.; BEHROO, Y. 2d hybrid chaos map for image security transform based on framelet and cellular automata. **Information Sciences**, Elsevier, v. 512, p. 855–879, 2020.

99 SONG, W. et al. A fast parallel batch image encryption algorithm using intrinsic properties of chaos. **Signal Processing: Image Communication**, Elsevier, p. 116628, 2022.

100 GHANE, A. H.; HARSINI, J. S. A network steganographic approach to overlay cognitive radio systems utilizing systematic coding. **Physical Communication**, Elsevier, v. 27, p. 63–73, 2018.

101 KANSO, A.; GHEBLEH, M. An algorithm for encryption of secret images into meaningful images. **Optics and lasers in engineering**, Elsevier, v. 90, p. 196–208, 2017.

102 PUSTOKHINA, I. V.; PUSTOKHIN, D. A.; SHANKAR, K. Blockchain-based secure data sharing scheme using image steganography and encryption techniques for telemedicine applications. In: **Wearable Telemedicine Technology for the Healthcare Industry**. [S.l.]: Elsevier, 2022. p. 97–108.

103 DEBNATH, B.; DAS, J. C.; DE, D. Design of image steganographic architecture using quantum-dot cellular automata for secure nanocommunication networks. **Nano communication networks**, Elsevier, v. 15, p. 41–58, 2018.

104 YANG, C.-N.; HSU, S.-C.; KIM, C. Improving stego image quality in image interpolation based data hiding. **Computer Standards & Interfaces**, Elsevier, v. 50, p. 209–215, 2017.

105 ZHOU, Y. et al. Information hiding scheme based on optical moiré-pixel matrix. **Optics Communications**, Elsevier, v. 437, p. 403–407, 2019.

106 MOHAMMED, H. A.; SAFFAR, N. F. H. A. Lsb based image steganography using mceliece cryptosystem. **Materials Today: Proceedings**, Elsevier, 2021.

107 LIANSHENG, S. et al. Multiple-image authentication based on the single-pixel correlated imaging and multiple-level wavelet transform. **Optics and Lasers in Engineering**, Elsevier, v. 130, p. 106102, 2020.

108 HALDER, S.; GHOSAL, A.; CONTI, M. Secure over-the-air software updates in connected vehicles: A survey. **Computer Networks**, Elsevier, v. 178, p. 107343, 2020.

109 YANG, Y.-G. et al. Using m-ary decomposition and virtual bits for visually meaningful image encryption. **Information Sciences**, Elsevier, v. 580, p. 174–201, 2021.

110 ____. Visually meaningful image encryption based on universal embedding model. **Information Sciences**, Elsevier, v. 562, p. 304–324, 2021.

111 SUBRAMANIAN, N. et al. A review of deep learning-based detection methods for covid-19. **Computers in Biology and Medicine**, Elsevier, p. 105233, 2022.

112 JAVED, A. R. et al. A comprehensive survey on digital video forensics: Taxonomy, challenges, and future directions. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 106, p. 104456, 2021.

113 YEDROUDJ, M.; COMBY, F.; CHAUMONT, M. Steganography using a 3-player game. **Journal of Visual Communication and Image Representation**, Elsevier, v. 72, p. 102910, 2020.

114 SAYAF, R.; PREIBUSCH, S.; CLARKE, D. Contextual healing: Privacy through interpretation management. In: IEEE. **2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)**. [S.l.], 2015. p. 360–365.

115 QIN, C. et al. Adversarial steganography based on sparse cover enhancement. **Journal of Visual Communication and Image Representation**, Elsevier, v. 80, p. 103325, 2021.

116 LI, Q. et al. An encrypted coverless information hiding method based on generative models. **Information Sciences**, Elsevier, v. 553, p. 19–30, 2021.

117 CHAUMONT, M. Deep learning in steganography and steganalysis. In: **Digital Media Steganography**. [S.l.]: Elsevier, 2020. p. 321–349.

118 ZHU, Z. et al. Destroying robust steganography in online social networks. **Information Sciences**, Elsevier, v. 581, p. 605–619, 2021.

119 LI, Q. et al. Image steganography based on style transfer and quaternion exponent moments. **Applied Soft Computing**, Elsevier, v. 110, p. 107618, 2021.

120 HAYES, J.; DANEZIS, G. Generating steganographic images via adversarial training. **arXiv preprint arXiv:1703.00371**, 2017.

121 WANG, Y.; FU, Z.; SUN, X. High visual quality image steganography based on encoder-decoder model. **Journal of Cybersecurity**, Tech Science Press, v. 2, n. 3, p. 115, 2020.

122 NAITO, H.; ZHAO, Q. A new steganography method based on generative adversarial networks. In: IEEE. **2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)**. [S.l.], 2019. p. 1–6.

123 BERNARD, S. et al. Explicit optimization of min max steganographic game. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 16, p. 812–823, 2020.

124 TABURET, T. et al. Natural steganography in jpeg domain with a linear development pipeline. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 16, p. 173–186, 2020.

125 WANG, Y.; CAO, Y.; ZHAO, X. Minimizing embedding impact for h. 264 steganography by progressive trellis coding. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 16, p. 333–345, 2020.

126 WU, J. et al. Audio steganography based on iterative adversarial attacks against convolutional neural networks. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 15, p. 2282–2294, 2020.

127 MAHATO, S.; YADAV, D. K.; KHAN, D. A. A minesweeper game-based steganography scheme. **Journal of Information Security and Applications**, Elsevier, v. 32, p. 1–14, 2017.

128 RAHIM, R.; NADEEM, S. et al. End-to-end trained cnn encoder-decoder networks for image steganography. In: **Proceedings of the European Conference on Computer Vision (ECCV) Workshops**. [S.l.: s.n.], 2018. p. 0–0.

129 BALUJA, S. Hiding images in plain sight: Deep steganography. **Advances in Neural Information Processing Systems**, v. 30, p. 2069–2079, 2017.

130 BOEHM, B. Stegexpose-a tool for detecting lsb steganography. **arXiv preprint arXiv:1410.6656**, 2014.

131 LE, Y.; YANG, X. Tiny imagenet visual recognition challenge. **CS 231N**, v. 7, n. 7, p. 3, 2015.

132 SHARMA, A. **Tiny ImageNet**. 2018. Disponível em: https://www.kaggle.com/datasets/akash2sharma/tiny-imagenet.

133 SAN, B. **Pokemon Mugshots**. 2019. Disponível em: https://www.kaggle.com/datasets/brilja/pokemon-mugshots-from-super-mystery-dungeon.