Instituto Federal de Brasília (IFB)-Campus Taguatinga

Curso: Bacharelado em Ciências da Computação

Matéria: Processamento Digital de Imagens

Professor: Raimundo Vasconcelos

Alunos: José Carlos de Jesus Santana Júnior

Como utilizar o algoritmo:

O algoritmo feito para estenografar uma imagem é o esteganografia.py utilizando a técnica de bit menos significativo (LSB), não tendo outros arquivos além desse. Para estenografar a imagem escolhida basta rodar o programa e digitar 1 caso se queira codificar a imagem, depois disso o programa pedirá para ser inserido o nome da imagem a ser codificada (já existente) junto com a extensão dela (exemplo: uva.png) sem aspas, logo em seguida, será mostrada a forma da imagem: altura, largura e canais de cor, respectivamente.O programa abrirá uma nova janela por meio do OpenCV para o usuário visualizar a imagem escolhida, feche essa janela e o usuário digitará a "mensagem secreta" a ser codificada em uma nova imagem. Para que isso ocorra o usuário deve digitar um nome de arquivo,com a extensão de arquivo, para a nova imagem com a mensagem secreta (exemplo uva2.png), por último o programa mostrará o número máximo de bytes possíveis de codificar, encerrando o programa. Depois de ter codificado uma imagem, ou ter uma imagem já previamente codificada de uma execução anterior do programa, para decodificar a mensagem dessa imagem é preciso executar o programa esteganografia.py novamente e dessa vez digitar 2 e logo em seguida digitar o nome da imagem codificada, com sua extensão (exemplo:uva1.png). A imagem codificada será mostrada em uma nova janela OpenCV, mas repare que no processo de estenografar não há nenhuma mudança aparente da imagem. Feche a janela OpenCV da imagem e o programa continuará executando e mostrará a mensagem codificada no terminal o que encerrará o programa.

(LSB)Estenografia de bit menos significativo:

O algoritmo feito para estenografar uma imagem é o esteganografia.py utilizando a técnica de bit menos significativo. Essa técnica funciona modificando o último bit de cada pixel para os bits da mensagem secreta. O algoritmo em questão altera os bits menos significativos dos pixels rgb (vermelho,verde e azul).Com essa técnica o impacto visual na imagem é mínimo, já que altera apenas um valor de cada pixel e esse é o mais a direita do byte, o que seria maior se fosse alterado os bits mais significativos, os mais à esquerda do byte. Como cada pixel contém valores de vermelho, verde ou azul variando de 0 a 255, ou seja cada um desses valores possuem 8 bits. Sendo assim, podemos converter a mensagem secreta, usando a tabela ASCII, em valores decimais e depois em binários. Em seguida basta interar/ir passando por cada valor do pixel, um de cada vez, e ir alterando o bit menos significativo do pixel por o bit da mensagem secreta que ainda não foi colocado até que a mensagem inteira esteja armazenada nos bits menos significativos. Por isso é importante saber quantos bits menos significativos temos na imagem para saber o tamanho máximo da mensagem que pode ser armazenada usando LSB.

Biblioteca OpenCV:

OpenCV é uma biblioteca multiplataforma com o intuito de ajudar o desenvolvimento de programas voltados para a área de Visão Computacional. Uma vez que é multiplataforma, é possível utilizar essa biblioteca com Python, Java e C++. Além disso, a biblioteca é open source e livre para uso acadêmico.

No algoritmo desenvolvido utilizamos OpenCV, mais especificamente a biblioteca cv2 que é o módulo para opencv-python. Para instalá-lo basta utilizar o comando a seguir em um terminal linux:

```
pip install opencv-contrib-python
```

Lembre-se de manter a versão do pip atualizada senão o comando não funcionará (a versão 19.3 é a versão mínima requerida). Para verificar a versão do pip, digite o comando:

```
pip -V
```

Para atualizar o pip, digite o comando:

```
pip install --upgrade pip
```

As funções utilizadas do cv2, que é o módulo das principais funcionalidades para Visão Computacional do OpenCV,foram:

- cv2.imshow(“nome da janela”,imagem_a_ser_mostrada): abre uma nova janela mostrando a imagem desejada com o nome desejado, para que funcione normalmente é preciso utilizar cv2.waitKey(0) e cv2.destroyAllWindows().
- cv2.waitKey(0): para a execução do programa e deixa a janela em aberto até que o usuário digite alguma coisa
- cv2.destroyAllWindows():responsável por fechar as janelas com as imagens corretamente sem “crashar” o programa, depois de fechada a janela o programa volta a executar normalmente.
- cv2.imread(nome_da_imagem): função que carrega a imagem do arquivo específico mencionado como parâmetro.Com essa função é possível trabalhar com a imagem como uma matriz. Exemplo: carrega as imagens desejadas no programa para que seja possível trabalhar com elas dentro do programa.

Todos os pacotes OpenCV utilizados são para Processamento Digitais de Imagens, gratuitos e para ambientes desktop padrão. Caso haja erros ou dúvidas a respeito da instalação visite o site: https://pypi.org/project/opencv-python/ (acesso em 17 de Novembro de 2021).

Biblioteca Numpy:

Numpy é uma biblioteca para Python que suporta o processamento de grandes, multidimensionais arranjos e matrizes oferecendo várias operações matemáticas para operar sobre esses dados. É uma popular biblioteca de ciências de dados para Python. Para instalá-la basta digitar o comando a seguir em um ambiente python:

```
pip install numpy
```

Caso haja erros ou dúvidas a respeito da instalação acesse o site: https://numpy.org/install/ (acesso em 17 de Novembro de 2021).

No algoritmo de esteganografia foi utilizado os seguintes módulos dessa biblioteca:

- numpy.ndarray(): cria um array em python (estrutura multidimensional, homogênea com um número fixo de itens), no algoritmo em questão foi utilizado caso a mensagem digitada pelo usuário seja desse tipo, seja possível convertê-la para binário.
  Documentação:https://numpy.org/doc/stable/reference/generated/numpy.ndarray.html (Acesso em 17 de Novembro de 2021)
- numpy.uint8(): tipo de dado que significa *8-bit unsigned integer*, Assim varia de 0 a 255.No algoritmo em questão é um dos tipos suportados que a mensagem secreta do usuário pode ser para conseguir convertê-la para binário.

Como funciona o algoritmo:

O programa é dividido em 6 funções:

- mensagemParaBinario(): responsável por pegar a mensagem original do usuário e convertê-la para um formato binário. Os formatos suportados de mensagem são: string, int,byte, nparray(array) e np.uint8(8-bits).Qualquer outro formato de mensagem ocorrerá em TypeError.
- esconderData(imagem, mensagem_secreta):tem como parâmetros a imagem original e a mensagem digitada pelo usuário e é responsável por esconder/codificar a mensagem na imagem usando LSB modificando os valores dos bits menos significativos vermelhos, verdes e azuis através de um laço. Essa função ocorrerá em ValueError se o tamanho da mensagem secreta for maior que o número de bits menos significativos da imagem. Antes disso ela chama a função mensagemParaBinario(mensagem). Retorna a imagem codificada.
- mostrarData(imagem): decodifica a mensagem da imagem codificada extraindo os dados dos bits menos significativos, divide por 8 bits e transforma em caracteres ASCII. Para ter certeza que o fim da mensagem

foi alcançado, é utilizado um delimitador “#####”: quando o delimitador é alcançado quer dizer que toda a mensagem foi convertida para caracteres ASCII e sai do laço de conversão. Retorna a mensagem decodificada menos o delimitador.

- codificado_texto():função chamada se o usuário digitou 1 para codificar a imagem, essa função pede o nome da imagem original e pede para o usuário digitar a mensagem secreta que ele queira inserir, chama a função esconderData() para codificar, e por último pede para o usuário digitar um novo nome para a imagem codificada e grava ela em disco, na mesma pasta que está o programa, usando OpenCV.
- decodificado_texto(): função chamada se o usuário digitou 2 para decodificar a imagem. Essa função pede o nome da imagem codificada, ou seja com uma mensagem secreta, mostra essa imagem, que geralmente não possui nenhuma diferença visual da original, chama a função mostrarData() para converter os binários menos significativos em caracteres, armazenando em uma variável e retorna a mensagem para que o usuário a veja no terminal.
- esteganografia(): serve como uma função principal para o programa, pergunta se o usuário quer codificar ou decodificar uma imagem, não podendo fazer as duas coisas em uma mesma execução, depois chama codificado_texto() se o usuário tiver digitado 1, ou chama decodificado_texto() se o usuário tiver digitado 2. O processo então de codificar e decodificar então é realizado pelas funções chamadas.

Testes e Limitações:

O programa foi testado com imagens png (*Portable Network Graphics*) principalmente, as principais imagens png não codificadas são: cachorro-filhote.png,ta-dando-onda.png e uva.png.Todas as imagens codificadas resultantes o nome é o mesmo da imagem original acrescentado de um número: por exemplo, as imagens codificadas de uva.png são uva1.png e uva2.png.Para essas imagens, as mensagens codificadas foram bem sucedidas por serem de um tamanho que não ultrapasse o limite de bits menos significativos(depende do tamanho da imagem).Quando a mensagem ultrapassa esse limite, é dado um erro ou o texto fica ilegível com caracteres estranhos. Além disso, foi notado que imagens com as extensões jpeg e jpg não mostraram a mensagem secreta corretamente ficando com caracteres ilegíveis. Outra coisa importante é que a extensão jpeg não reconhece o atributo imagem.shape[0] do código o que faz que o código dê uma falha. Um exemplo de mensagem secreta maior que a quantidade de bits disponíveis é a uva3.png, e um teste para ver quais caracteres são suportados na mensagem é o cachorro-filhote2.png. Não foi testado o programa com todos os caracteres Unicode, mas os caracteres não pertencentes à tabela ASCII, a mensagem não é mostrada para o usuário corretamente ficando com texto ilegível (exemplo:cachorro-filhote3.png).Caso se queira codificar uma imagem já codificada, a codificação ocorre sem problemas, mesmo que a mensagem original seja maior que a mensagem que será escrita no momento(para pngs). Além disso, se as imagens não estiverem na mesma

pasta que o programa será preciso informar o caminho dessas ao escrever o nome do arquivo para codificar.

Referências:

1. https://towardsdatascience.com/steganography-hiding-an-image-inside-another-77ca66b2acb1
2. https://www.edureka.co/blog/steganography-tutorial
3. https://www.thepythoncode.com/article/hide-secret-data-in-images-using-steganography-python
4. https://www.youtube.com/watch?v=xepNoHgNjow&t=1922s