

MC920 - Trabalho 2

Nome: Bruno Orsi Berton

RA: 150573

Versões usadas	1
Como executar	1
Saída do programa	2
Análise do programa	2
Limitações	5

Versões usadas

Python - 3.5.2

NumPy - numpy-1.14.1

Scipy - 1.0.0

Como executar

Para executar o programa que faz a codificação, basta executar o comando:

python3 codificar.py <caminho relativo para a imagem de entrada> <caminho relativo para o texto de entrada> <plano de bits> <imagem de saída>

Para executar o programa que faz a decodificação, basta executar o comando:

python3 decodificar.py <caminho relativo para a imagem que contém a mensagem> <plano de bits> <arquivo texto para a mensagem ser armazenada>

Exemplo:

Para codificar:

**python3 codificar.py ../images/baboon.png ../text/input-text.txt 0
output-image.png**

Para decodificar:

python3 decodificar.py output-image.png 0 output-text.txt

Saída do programa

- O programa **codificar.py** gera a imagem com o texto codificado em seus bits.

- O programa **decodificar.py** gera o arquivo texto de saída com a mensagem codificada no programa anterior.

Análise do programa

Para codificar uma mensagem qualquer em uma imagem, primeiro é recuperado o código ASCII de cada caractere da mensagem e esse código é convertido para binário.

Agora com o texto convertido para binário, cada banda de cor de cada pixel da imagem pode receber exatamente um bit da mensagem, então os valores de vermelho(R), verde(G) e azul(B) são convertidos para binário também, apenas um bit de cada um é alterado, então essa binário é convertido de volta para inteiros, e a imagem de saída é salva.

Para que seja possível identificar o final da mensagem, é adicionado um texto logo após a mensagem do usuário. “[#END#]” identifica que a mensagem acabou, e que o resto é apenas lixo sem sentido.

Para exemplificar, codificamos as 3 leis da robótica de Isaac Asimov na imagem **monalisa.png** utilizando o plano de bits **2**, que altera o 3 bits menos significativo de cada banda de cor de cada pixel.

Mensagem:

1ª lei: Um robô não pode ferir um ser humano ou, por inacção, permitir que um ser humano sofra algum mal.

2ª lei: Um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei.

3ª lei: Um robô deve proteger sua própria existência, desde que tal proteção não entre em conflito com a Primeira e Segunda Leis.

Imagem antes da codificação



Imagem após a codificação:



E o texto recuperado pela programa de decodificar:

1ª lei: Um robô não pode ferir um ser humano ou, por inacção, permitir que um ser humano sofra algum mal.

2ª lei: Um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei.

3ª lei: Um robô deve proteger sua própria existência, desde que tal proteção não entre em conflito com a Primeira e Segunda Leis.

Como podemos ver, não é possível ver nenhum ruído gerado pelo processo de codificação nesta imagem e a mensagem foi recuperada com perfeição. Para validar se a imagem continua sem ruído, vamos encodar o máximo de caracteres que a imagem conseguir, no plano de bits 2 novamente. Para isso, foi codificado vários parágrafos cheios do caractere “z”, que é a letra do alfabeto com o maior valor na tabela ASCII, 122.

Imagem antes da codificação:

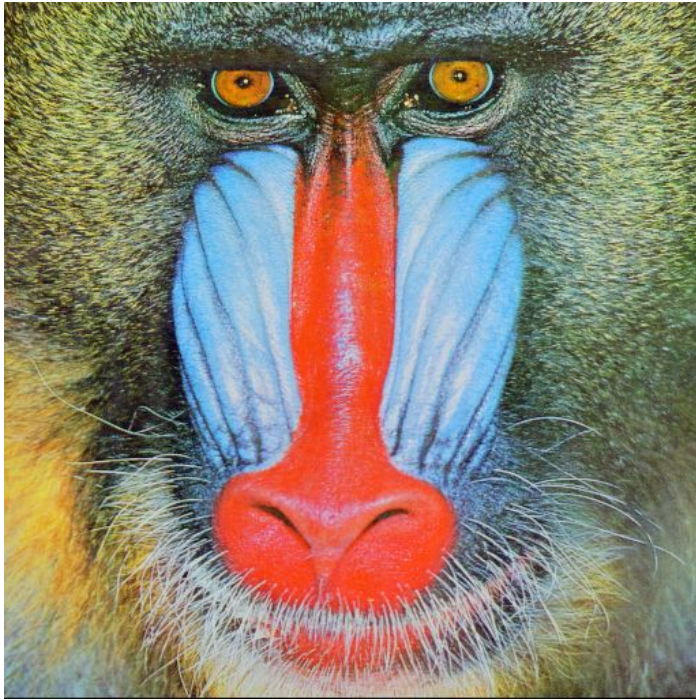
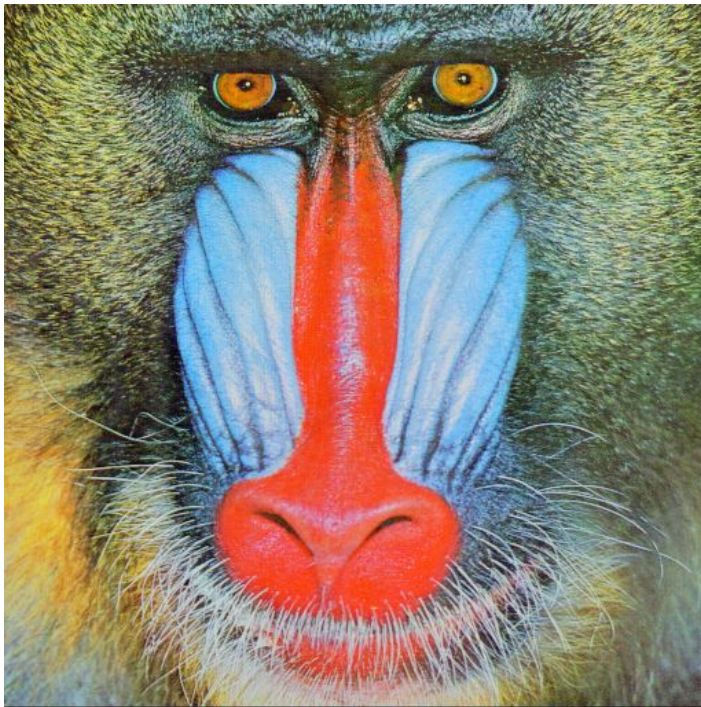


Imagem depois da codificação:



De novo, é imperceptível a presença de ruído na imagem ao olho humano.

Limitações

Cada imagem permite que um certo tamanho de texto seja armazenado. E pode ser calculado como $\text{largura} * \text{altura} * 8 \text{ bits}$. No caso, usamos 7 caracteres para armazenar o que indica o fim da mensagem, então temos que a quantidade máxima de caracteres que podem ser armazenados nas imagens são $\text{largura} * \text{altura} * 8 \text{ bits} + 7 * 8$, qualquer mensagem que ultrapasse esse limite, não será codificada na imagem.