

**Universidade Estadual de Campinas
Instituto de Computação**

Introdução ao Processamento Digital de Imagem (MC920 / MO443)

Professor: Hélio Pedrini

Aluno: Gabriel Pedroso Mariani

Ra: 197470

Trabalho 4

1. O problema

Nosso objetivo neste trabalho é armazenar informações ocultas em imagens, esta técnica é chamada estenografia. Existem diversos usos práticos para tal técnica como verificação de copyright.

A técnica escolhida para este trabalho, foi a alteração dos bits menos significativos de cada cor de cada pixel para que pudesse armazenar a informação. Quanto menos significativo for o bit menor é a alteração na imagem, mas quanto mais bits se usar mais informações podem ser guardadas.

Essa técnica geralmente não gera alterações visíveis a olho nu em uma imagem, por isso pode se fatiar a imagem em planos de bits para que as alterações fiquem mais claras.

2. Modo de uso do Script

-Codificar

Esse Script faz uso da biblioteca OpenCv, NumPy, Math e para rodar basta definir 3 parâmetros.

-i (Requerido) - Caminho para a imagem a ser usada.

-d (Requerido) - Nome e formato da imagem destino.

-m (Requerido) - Caminho para a mensagem a ser ocultada.

-b - Quantidade de bits significativos a serem usados (0 (default), 1 ou 2).

Ex. python codificar.py -i baboon.png -d baboon_coded.png -m msg.txt -b 1

-Decodificar

Esse Script faz uso da biblioteca OpenCv, NumPy, Math e para rodar basta definir 2 parâmetros.

-i (Requerido) - Caminho para a imagem a ser usada.

-d (Requerido) - Nome e formato da mensagem destino.

-b - Quais planos de bits foram usados na codificação (0 (default), 1 ou 2).

Ex. *python decodificar.py -i baboon_coded.png -d mensagem_saida.txt -b 1*

Obs: Os arquivos de mensagem precisam estar na codificação ASCII ou ANSI.

3. Modo de implementação

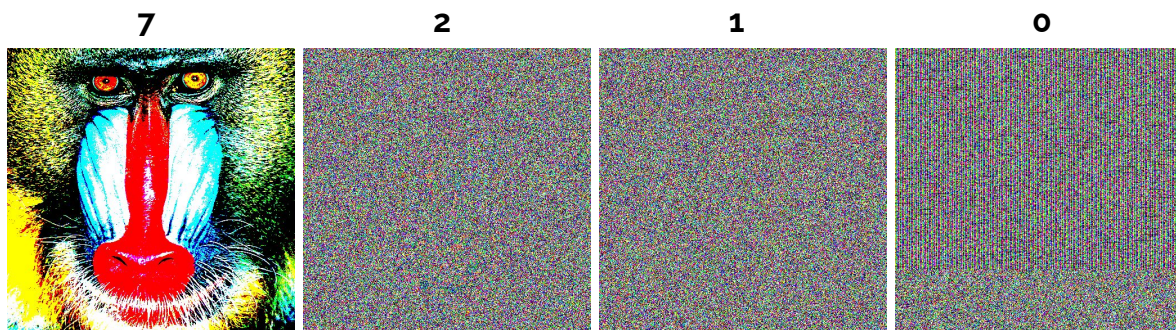
O código não fez uso de biblioteca com métodos prontos, apenas usou as ferramentas básicas como operações binárias.

O script codificar oculta uma dada mensagem em uma dada imagem. Se o bitplan escolhido for zero, apenas é usado o bit menos significativo, se for escolhido 1, o primeiro e o segundo serão usados, e se for escolhido 2, os 3 menos significativos serão usados. A mensagem é codificada ordenadamente pelos bits e dentro de cada bit na ordem de azul, verde e vermelho (BGR). A condição de parada é o caractere 255 ascii que é adicionado ao fim da mensagem, caso a mensagem contenha esse caráter, o script decodificar não funcionará apropriadamente. Se a mensagem for maior do que a imagem comporta, então o script codificará até a imagem acabar.

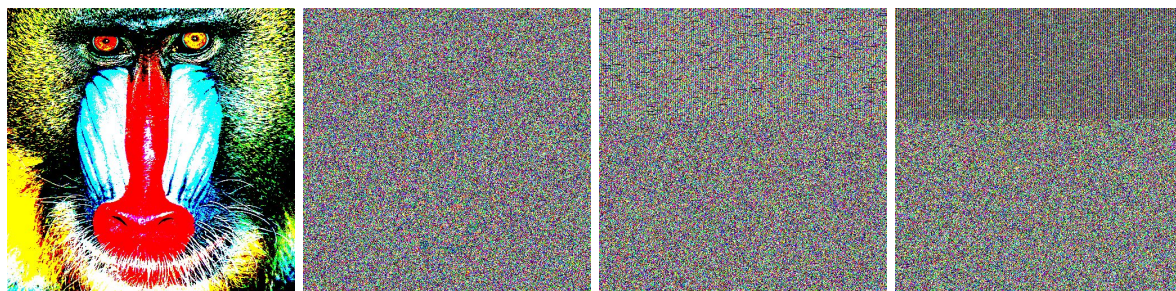
O script decodificar lê uma mensagem oculta dada uma imagem. É necessário saber quais planos de bits foram usados na codificação para a decodificação. O critério de parada é o caractere 255 ascii. Caso não encontre o caractere 255 o script decodificará até o fim da imagem. A decodificação segue a mesma ordem da codificação.

4. Resultados

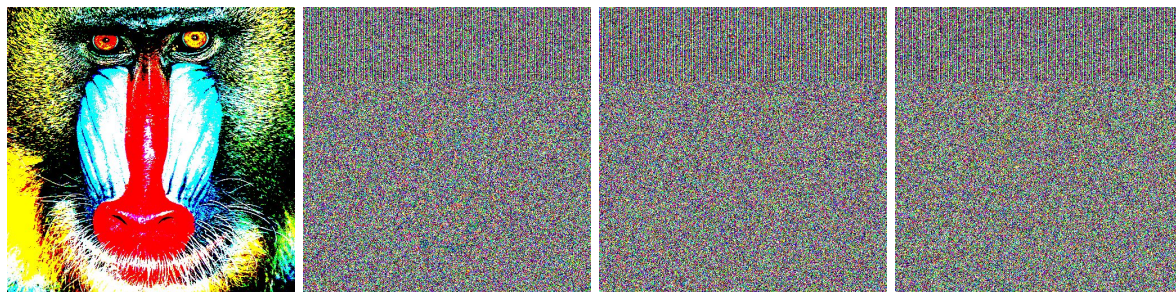
BitPlan = 0



BitPlan = 1



BitPlan = 2



5. Conclusões

Analisando os resultados, pode-se perceber o surgimento de um padrão no plano de bits que foi usado para armazenar a mensagem. Também pode-se perceber que quanto mais bits se usa mais espaço para mensagem se tem. Além disso, conclui-se que a estenografia nos 3 primeiros bits é praticamente imperceptível aos olhos humanos.

6. Considerações finais

O aparecimento de padrões que possibilita descobrir se uma imagem carrega ou não uma mensagem não é um problema, já que o foco da estenografia não é encriptar dados.

O trabalho possibilitou um conhecimento mais incisivo sobre técnicas de ocultação de informações em imagens e possibilitou a utilização de muitas operações binárias.

O código de extração de planos não foi incluído no envio dos arquivos do trabalho.