

# Escondendo uma mensagem em uma imagem (trabalho 1)

## Introdução ao Processamento de Imagem Digital

Randerson A. Lemos (103897) 2022-1S

## 1 Introdução

A esteganografia é a área de conhecimento que se dedica a estudar técnicas de ocultação de informações (por exemplo, mensagens) em imagens sem que estas sofram alterações perceptíveis ao olho humano. A esteganografia, neste trabalho, será aplicada por meio da substituição dos bits menos significativos dos pixels das imagens escolhidas pelos bits da mensagem de interesse que se deseja ocultar na imagem. A seguir teremos três seções, a de Solução, a de Resultado e a de Conclusão. Na seção de Solução, detalhes técnicos, de usabilidade e de decisões da solução proposta são fornecidos. Na seção de Resultado, os principais resultados são apresentados. Na seção de Conclusão, há não apenas a apresentação de conclusões, mas também de discussões.

## 2 Solução

A solução utiliza a linguagem de programação Python e conta com o auxílio do gerenciador de projetos e pacotes Conda. Assumindo que o usuário tenha o Conda instalado em sua máquina, a configuração do projeto pode ser feita pela execução do comando `conda env create -f environment.yml` a partir da pasta **trab1**. Esse comando cria o ambiente de trabalho **mc920-trab1** e instala os seguintes módulos: `opencv`, `numpy`, `scipy`, `pandas`, `matplotlib`. Finalizada a configuração do ambiente de trabalho **mc920-trab1**, o usuário deve executar o comando `source source.sh`<sup>1</sup> para carregar as variáveis de ambiente adequadas e, assim, poder usar os programas do projeto dentro do ambiente de trabalho recém configurado.

Três programas foram criados, os quais são: **codificar.py**, **decodificar.py**, **mostrar\_planos.py**. Na sequência, as funcionalidades de cada um desses programas é detalhada.

### 2.1 Codificar.py

O programa **codificar.py** é responsável por ocultar uma mensagem de interesse em uma imagem escolhida que deve ser colorida e estar no formato png. Para ser executado, esse programa precisa receber os parâmetros **imagem\_entrada**, **texto\_entrada**, e **planos\_bits**:

- ao parâmetro **imagem\_entrada** deve-se fornecer o nome do arquivo da imagem a ser utilizada como ‘bau’ da mensagem que se deseja esconder;
- ao parâmetro **texto\_entrada** deve-se fornecer o nome do arquivo do texto que contém a mensagem que se deseja ocultar;
- ao parâmetro **planos\_bits** deve-se fornecer os planos de bits menos significativos dos pixels da imagem escolhida que serão utilizados para registrar a mensagem de interesse. Os valores esperados para esse parâmetro são: 0, ou 1, ou 2, ou combinações desses valores separados por ‘:’. Quando mais de um plano de bits são passados ao programa pelo parâmetro **planos\_bits**, ocorre a ordenação em ordem crescente desses planos de modo que a utilização dos bits dos pixels da imagem se dê sempre do plano de bits menos significativo para o mais significativo.

Alguns exemplos de como utilizar o programa **codificar.py** são: `python3 codificar.py -imagem_entrada=png/watch.png -texto_entrada=txt/texto1.txt -planos_bits=2`; `python3 codificar.py -imagem_entrada=png/watch.png -texto_entrada=txt/texto1.txt -planos_bits=1:2`; `python3 codificar.py -imagem_entrada=png/watch.png -texto_entrada=txt/texto1.txt -planos_bits=0:1:2`

### 2.2 Mostrar\_planos.py

```
python3 mostrar_planos.py -imagem_entrada=out/watchm_plano12.png -planos_bits=0:1:2;
```

---

<sup>1</sup>O comando que configura o ambiente de trabalho **mc920-trab1** precisa ser executado apenas um vez. Assim sendo, depois que este ambiente está configurado, o usuário precisa apenas executar o comando `source source.sh`