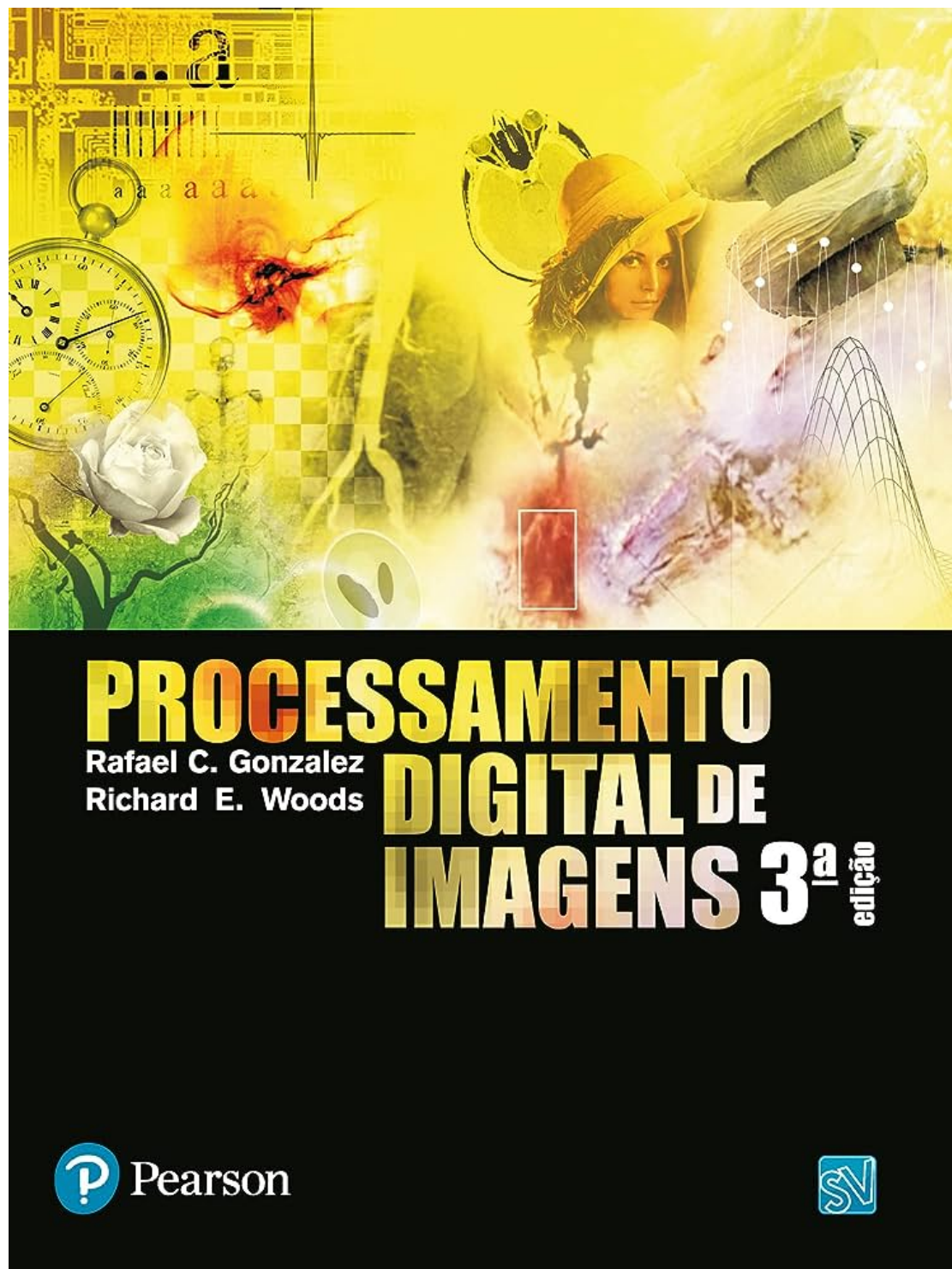


# PDIE8 - Processamento Digital de Imagem



Repositorio dos Códigos feitos em aula da Disciplina

Feito por: Vinicius de Souza Santos

Lecionado por: Murilo Varges

## Aula 5 - 28/08 - Transformações de Intensidade

- Transformações de intensidade
- Fundamentos
- Histograma
- Equalização de histograma

### Exercícios - Transformações e Histograma

- Utilizar as imagens Fig 3.8 e enhance-me.gif disponíveis no Moodle
- Aplicar a transformação logarítmica, testar vários valores para o parâmetro  $c$  " $s = c \log(1 + r)$ "
- Aplicar a transformação de potência (gama), testar vários valores para o parâmetro  $\gamma$  e  $c=1$  " $s = cr^\gamma$ "
- Implemente a representação de cada plano de bits das imagens
- Implementar a equalização do histograma
- Elaborar relatório explicando a implementação de cada transformação e qual foi o efeito na imagem.

Importando Bibliotecas

```
In [ ]: import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
```

Funções de Transformação

```
In [ ]: def log_transform(c, img):
    return c * np.log(1 + img)

def power_transform(c, gamma, img):
    return c * np.power(img, gamma)

def bit_plane(img, bit):
    return (img & (1 << bit)) >> bit
```

Leitura e Processamento das Imagens

```
In [ ]: enhance_path = '/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/
Fig0308_path = '/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem

# Leitura das imagens
enhance_img = np.array(Image.open(enhance_path))
Fig0308_img = np.array(Image.open(Fig0308_path))

# Aplicação das transformações
c_values = [1, 5, 10, 20]
gamma_values = [0.1, 0.5, 1, 2, 5]
```

```

for c in c_values:
    # Transformação Logarítmica
    transformed_enhance = log_transform(c, enhance_img)
    transformed_Fig0308 = log_transform(c, Fig0308_img)

    # Exibição
    fig, axs = plt.subplots(1, 2, figsize=(10, 5))
    axs[0].imshow(transformed_enhance, cmap='gray')
    axs[0].set_title(f'Enhance-me (c = {c})')
    axs[1].imshow(transformed_Fig0308, cmap='gray')
    axs[1].set_title(f'Fig0308 (c = {c})')
    plt.show()

for gamma in gamma_values:
    # Transformação de Potência (Gama)
    transformed_enhance = power_transform(1, gamma, enhance_img)
    transformed_Fig0308 = power_transform(1, gamma, Fig0308_img)

    # Exibição
    fig, axs = plt.subplots(1, 2, figsize=(10, 5))
    axs[0].imshow(transformed_enhance, cmap='gray')
    axs[0].set_title(f'Enhance-me (γ = {gamma})')
    axs[1].imshow(transformed_Fig0308, cmap='gray')
    axs[1].set_title(f'Fig0308 (γ = {gamma})')
    plt.show()

# Representação de cada plano de bits
for i in range(8):
    bit_enhance = bit_plane(enhance_img, i)
    bit_Fig0308 = bit_plane(Fig0308_img, i)

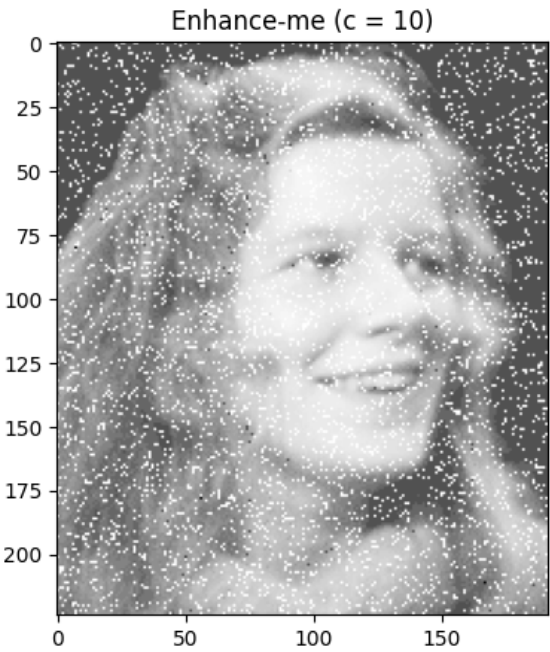
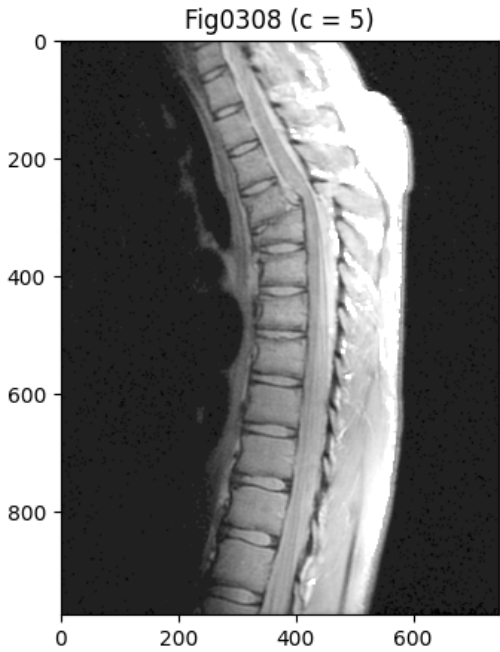
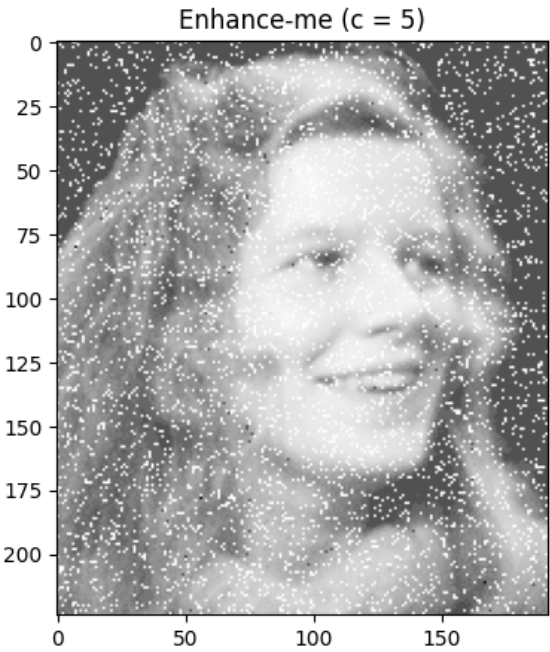
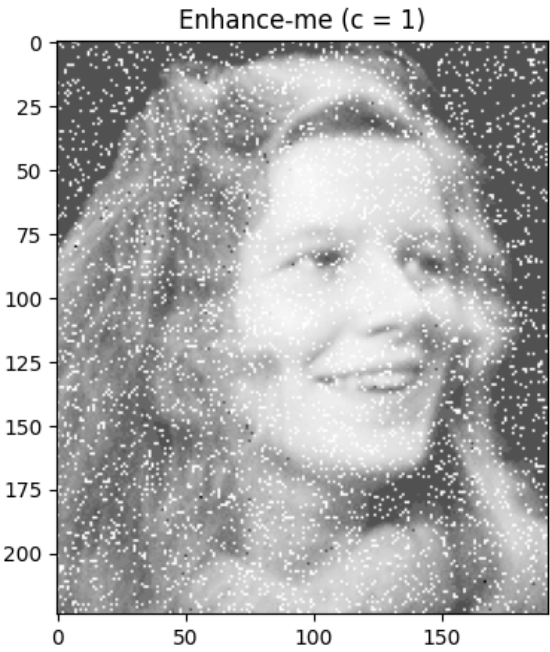
    # Exibição
    fig, axs = plt.subplots(1, 2, figsize=(10, 5))
    axs[0].imshow(bit_enhance, cmap='gray')
    axs[0].set_title(f'Enhance-me (Bit Plane {i})')
    axs[1].imshow(bit_Fig0308, cmap='gray')
    axs[1].set_title(f'Fig0308 (Bit Plane {i})')
    plt.show()

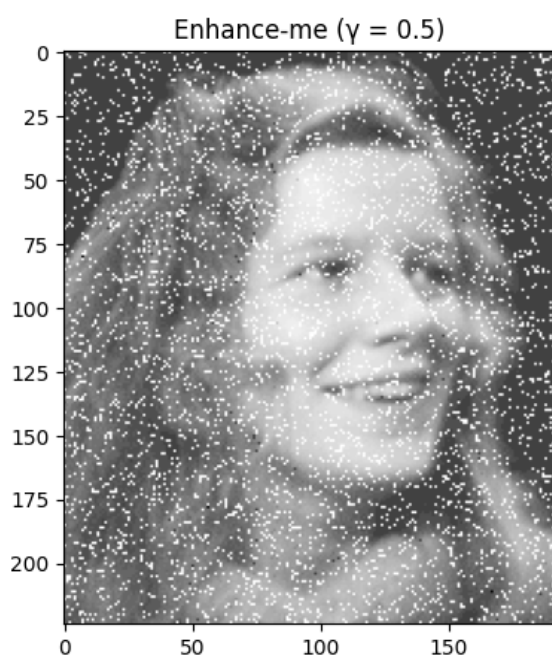
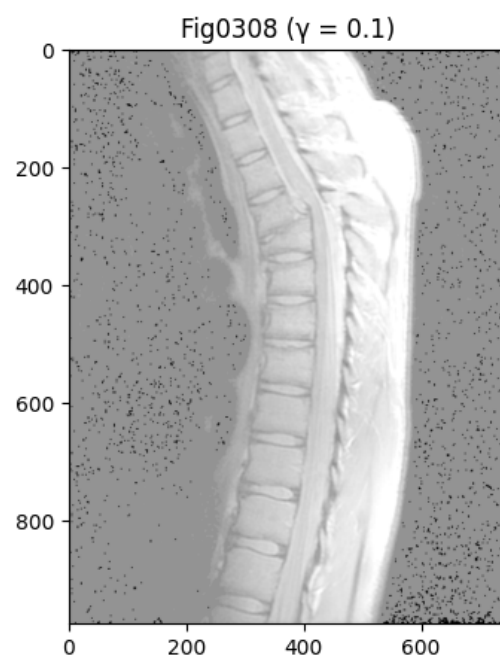
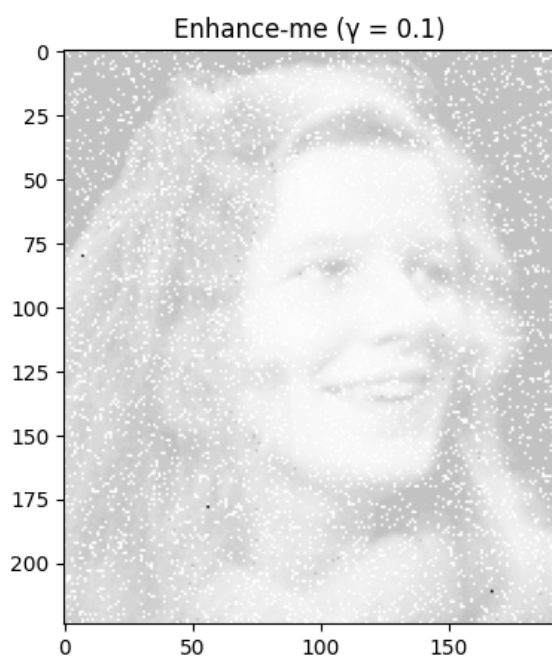
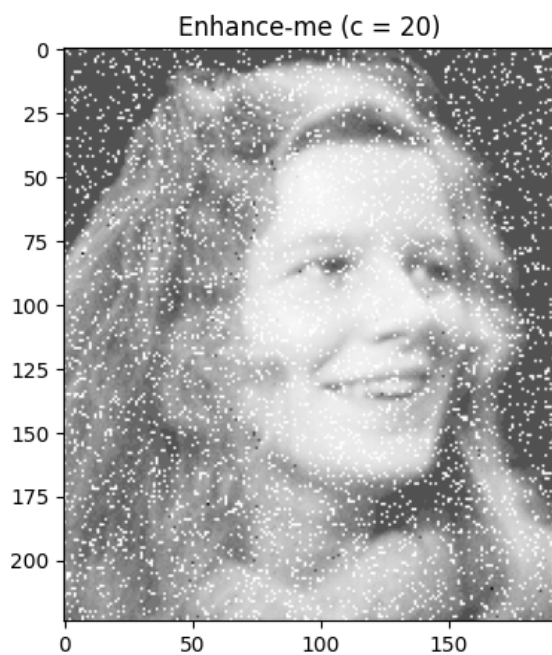
# Equalização do histograma
equalized_enhance = cv2.equalizeHist(enhance_img)
equalized_Fig0308 = cv2.equalizeHist(Fig0308_img)

# Exibição
fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].imshow(equalized_enhance, cmap='gray')
axs[0].set_title('Enhance-me (Equalização)')
axs[1].imshow(equalized_Fig0308, cmap='gray')
axs[1].set_title('Fig0308 (Equalização)')
plt.show()

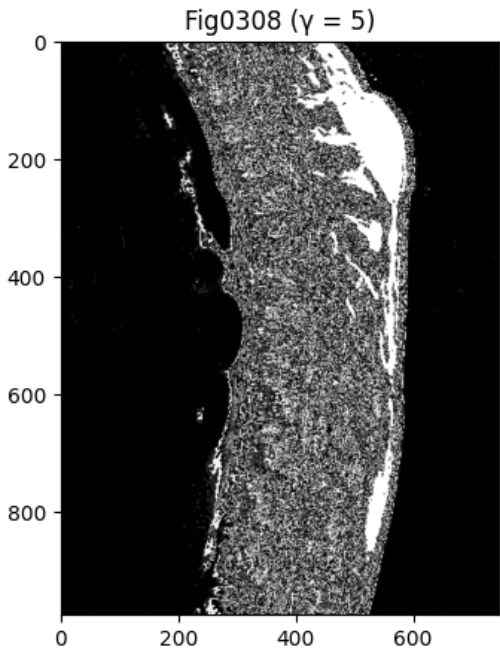
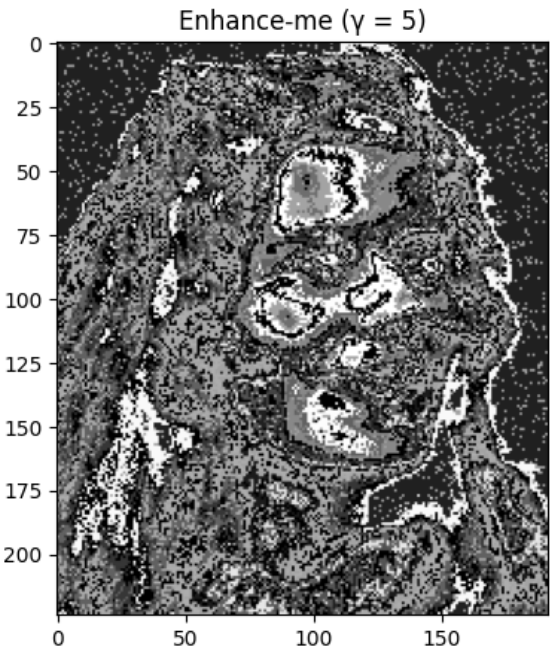
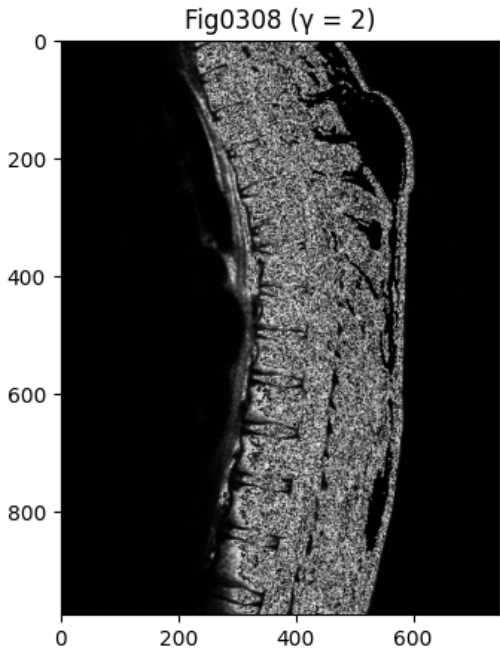
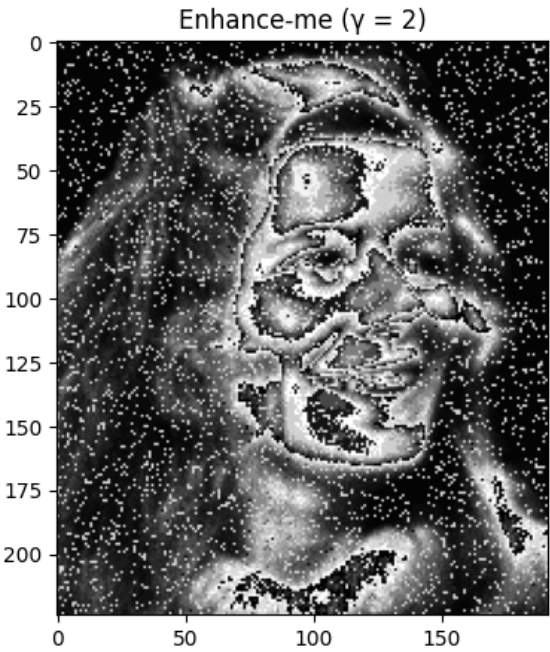
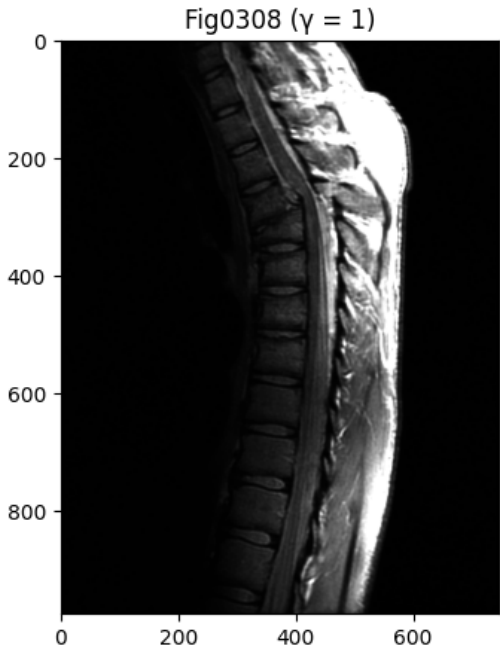
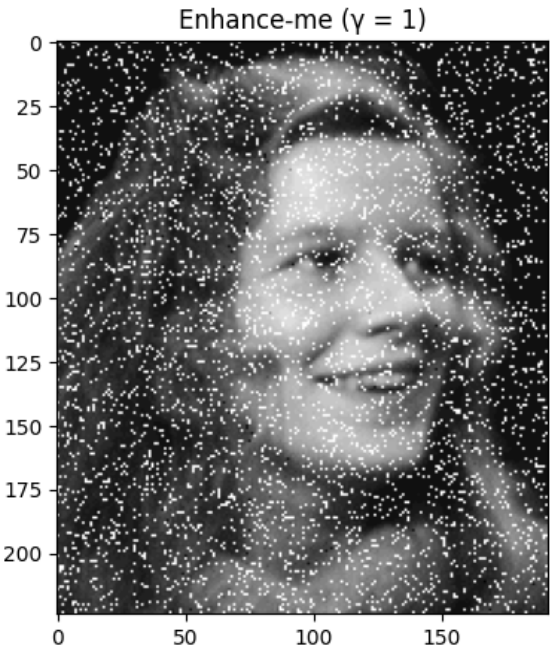
```

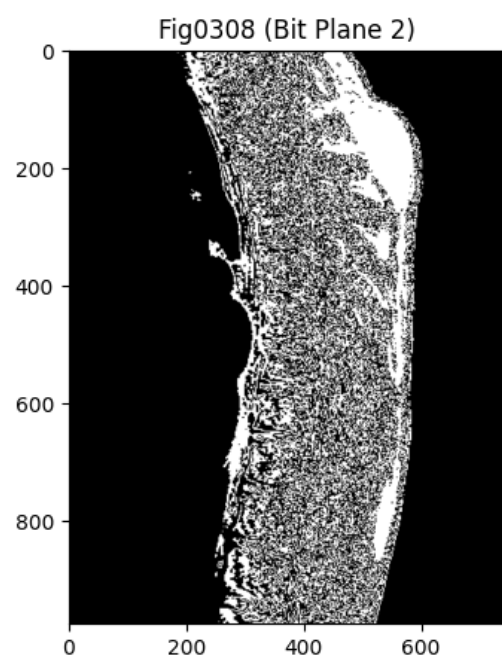
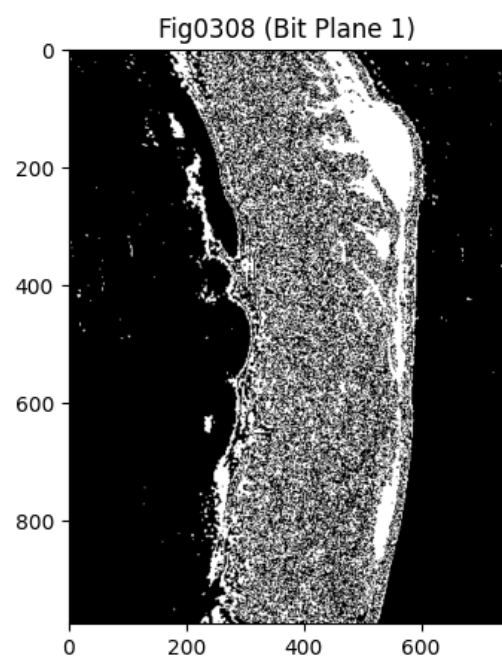
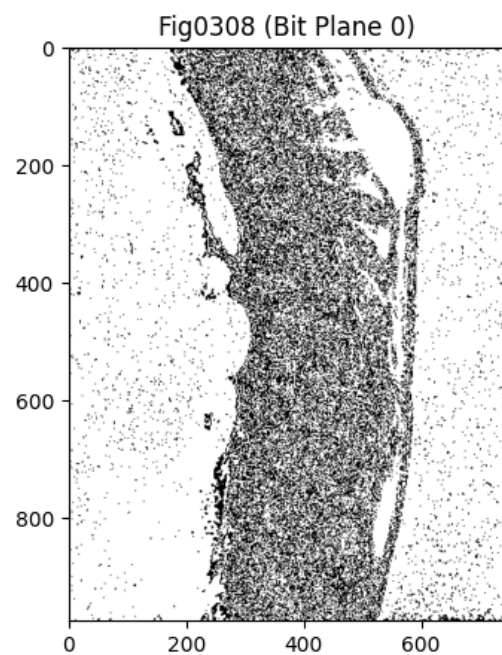
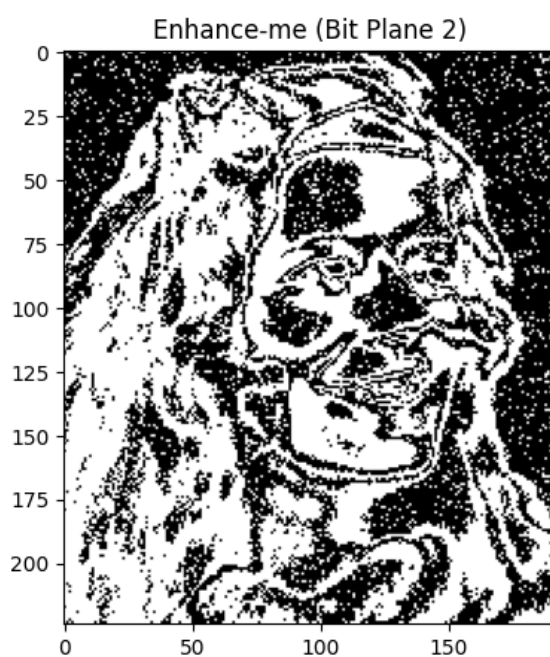
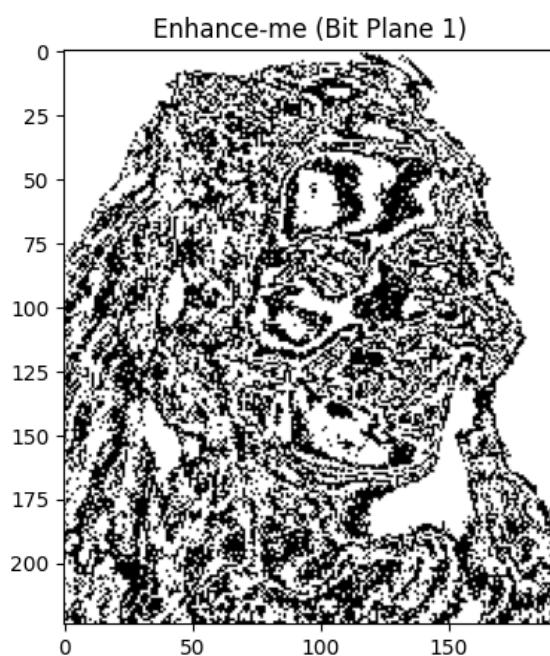
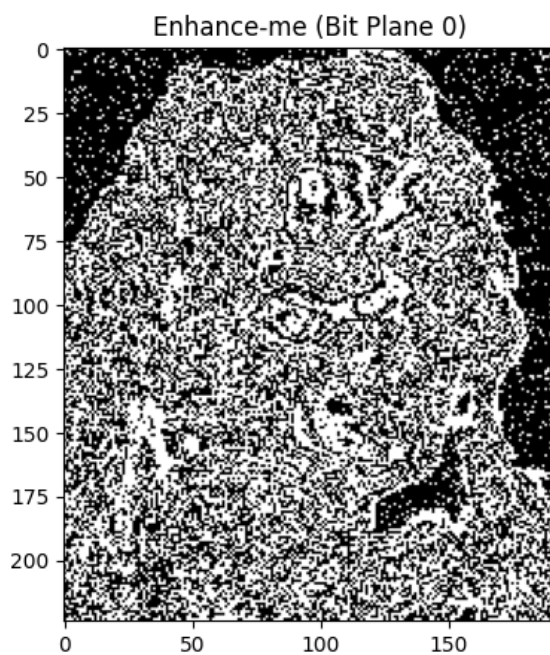
C:\Users\vinny\AppData\Local\Temp\ipykernel\_20780\382409465.py:2: RuntimeWarning:  
 divide by zero encountered in log  
 return c \* np.log(1 + img)

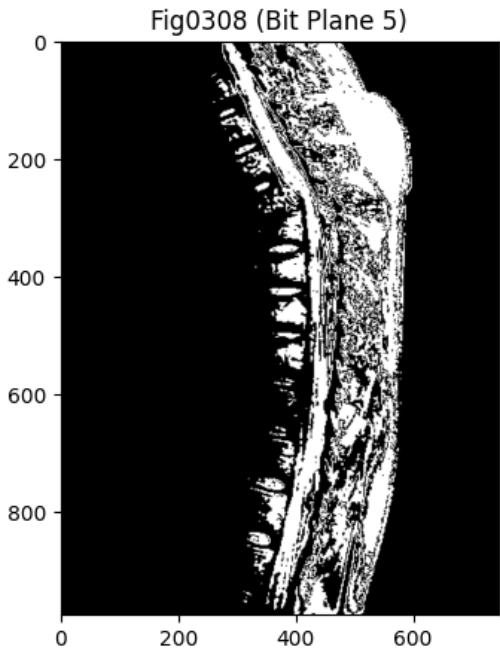
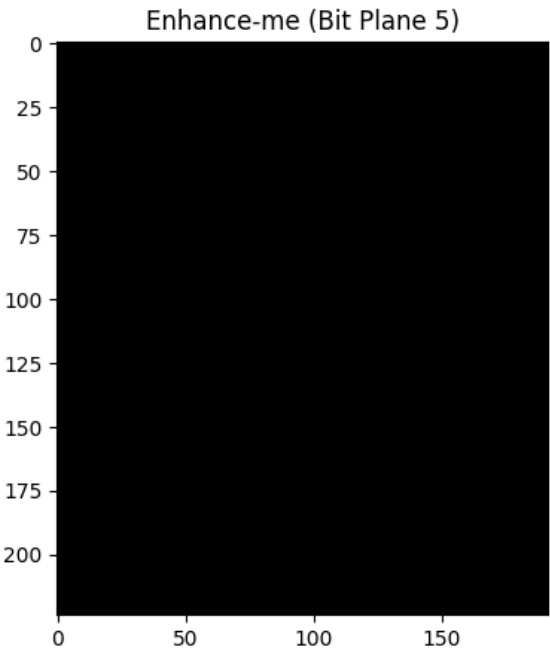
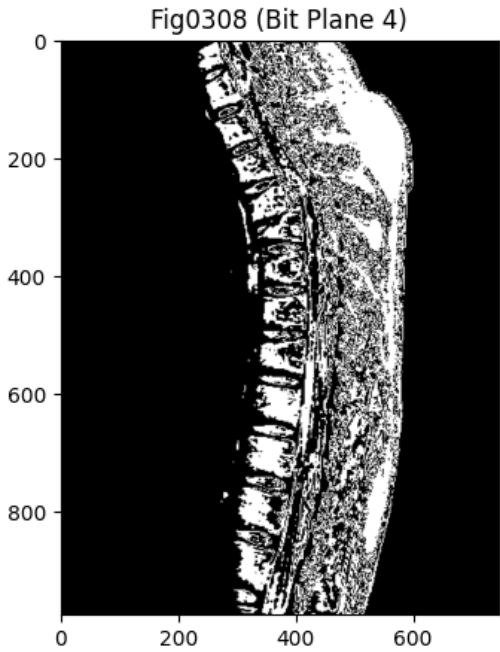
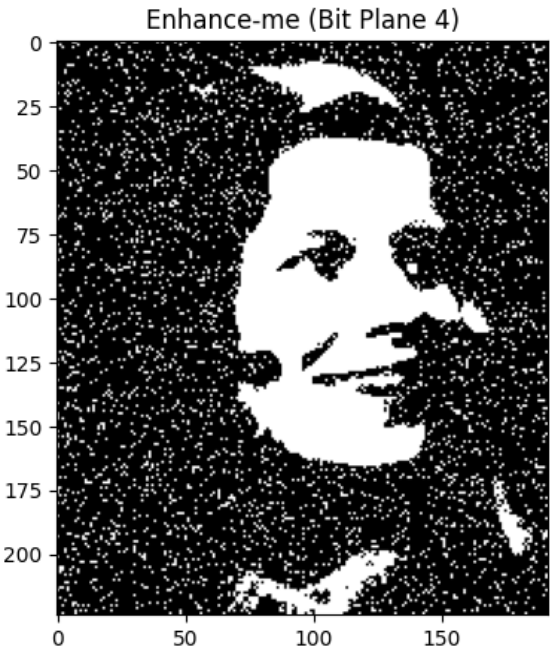
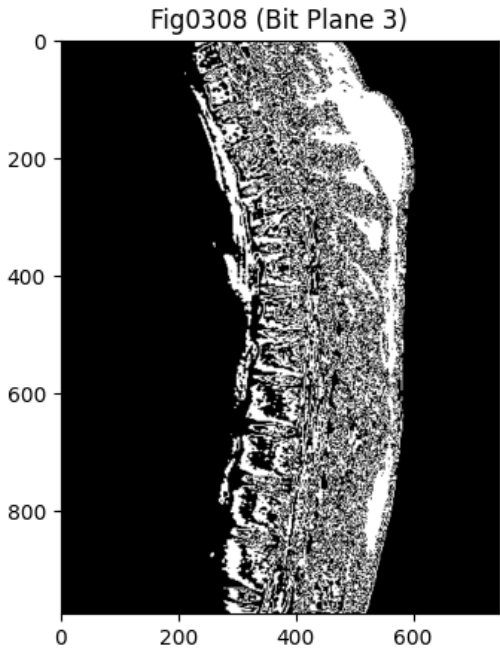
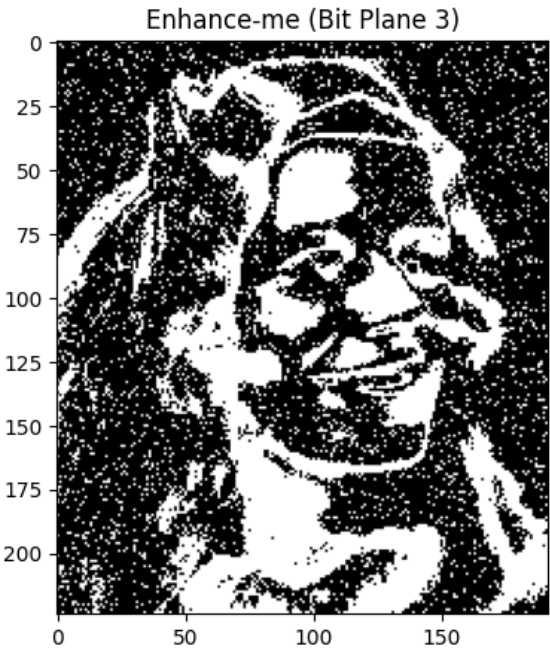




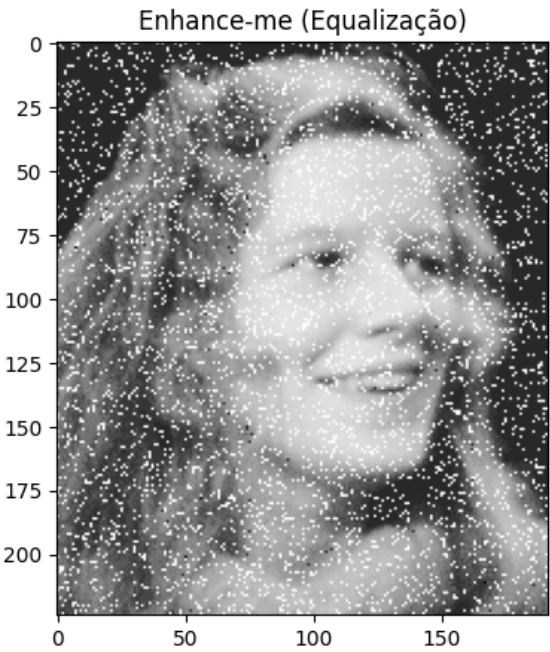
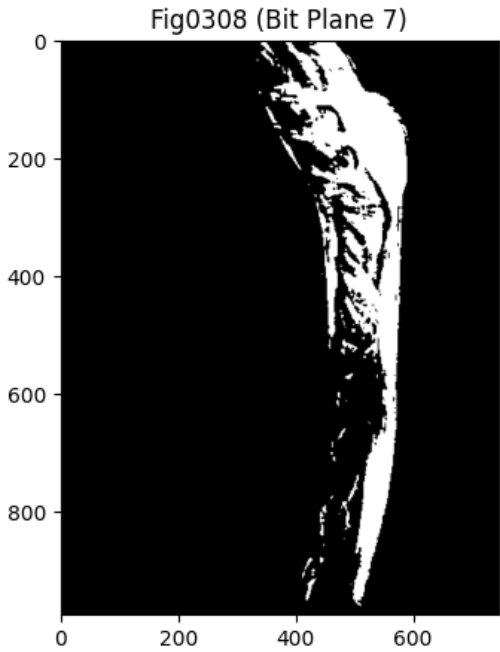
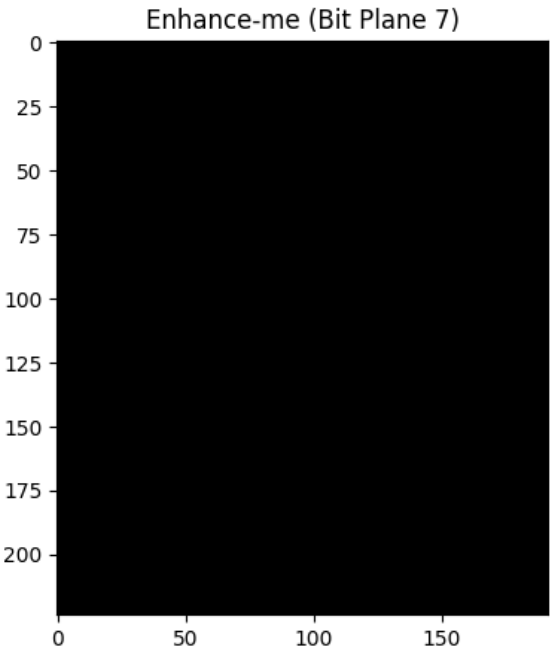
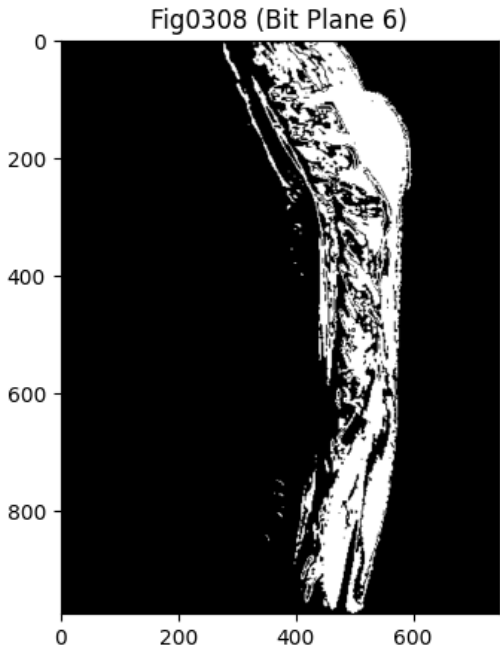
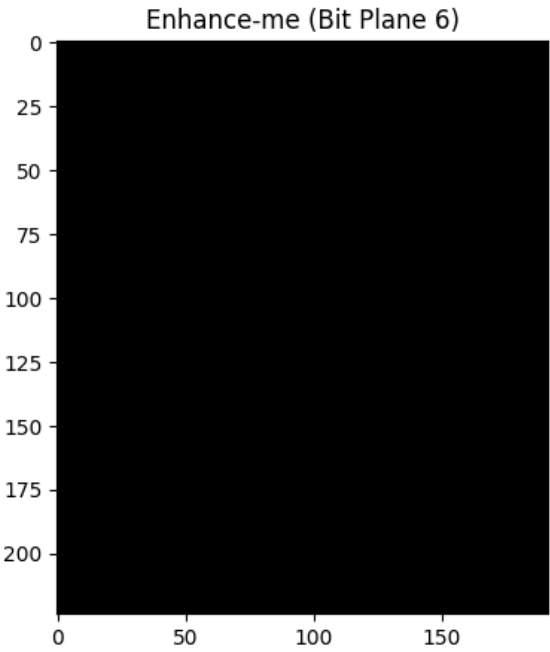












# Relatório: Análise e Implementação de Transformações em Processamento Digital de Imagem

## 1. Introdução

O processamento digital de imagens (PDI) é uma área que se concentra na manipulação e análise de imagens digitais. Este relatório detalha a aplicação de quatro técnicas de PDI em duas imagens distintas: enhance-me.gif e Fig0308(a)(fractured\_spine).tif. As técnicas abordadas são: transformação logarítmica, transformação de potência (gama), representação de plano de bits e equalização de histograma.

## 2. Transformação Logarítmica

### 2.1 Fundamentação Teórica

A transformação logarítmica visa expandir a escala de valores de pixels escuros e comprimir a de pixels claros. É útil para imagens com grandes variações de brilho.

### 2.2 Implementação

A transformação é aplicada através da fórmula:

$s = c \times \log(1 + r)$  onde  $r$  é o valor do pixel original e  $c$  é uma constante.

### 2.3 Efeito nas Imagens

Ao variar o valor de  $c$ , observa-se uma alteração no contraste das imagens. Valores maiores de  $c$  resultam em um contraste mais acentuado, enquanto valores menores suavizam o efeito. Esta transformação é especialmente útil para realçar detalhes em regiões escuras das imagens.

## 3. Transformação de Potência (Gama)

### 3.1 Fundamentação Teórica

A transformação de potência, ou correção gama, é utilizada para ajustar o brilho global de uma imagem.

### 3.2 Implementação

A transformação é dada pela fórmula:

$s = c \times r^\gamma$ , onde  $\gamma$  é o valor que determina o grau de correção.

### 3.3 Efeito nas Imagens

Valores de  $\gamma < 1$  clareiam a imagem, tornando-a mais luminosa, enquanto valores de  $\gamma > 1$  escurecem a imagem. Esta transformação é útil para ajustar imagens capturadas sob diferentes condições de iluminação.

## 4. Representação de Cada Plano de Bits

### 4.1 Fundamentação Teórica

Cada pixel de uma imagem em escala de cinza pode ser representado em termos de seus bits individuais.

### 4.2 Implementação

A representação de cada plano de bits é obtida isolando-se cada bit do valor do pixel.

### 4.3 Efeito nas Imagens

O plano de bit mais significativo contém a maior parte da informação visual. Planos de bits inferiores contribuem com detalhes mais sutis e, frequentemente, com ruído. Esta técnica é útil para analisar a contribuição de cada bit para a imagem global.

## 5. Equalização do Histograma

### 5.1 Fundamentação Teórica

A equalização de histograma redistribui os valores de pixel de uma imagem para produzir um histograma uniforme.

### 5.2 Implementação

A equalização é realizada através da função `cv2.equalizeHist()`.

### 5.3 Efeito nas Imagens

A técnica melhora o contraste global das imagens, tornando os detalhes mais visíveis e distribuindo a intensidade dos pixels de forma mais uniforme.

## 6. Conclusão

As técnicas de processamento digital de imagem discutidas neste relatório são essenciais para a manipulação e melhoria de imagens digitais. A aplicação destas técnicas em duas imagens distintas demonstrou sua eficácia e versatilidade. A compreensão teórica, juntamente com a implementação prática, fornece uma base sólida para explorar aplicações mais avançadas no campo do PDI.