

PDIE8 - Processamento Digital de Imagem

Feito por: Giovana Perazzolo Menato e Vinicius de Souza Santos

Lecionado por: Murilo Varges

Aula 4 - 21/08 - Exercícios

- Resolução de exercícios sobre fundamentos de imagens digitais;
- Operações ponto a ponto;
- Operações por vizinhança;
- Transformações geométricas.

Alunos: Giovana Menato e Vinicius Santos

Exercicio 1

[OPERAÇÃO PONTO A PONTO]:

- Calcular o negativo das imagens;
- Diminuir pela metade a intensidade dos pixels;
- Incluir 4 quadrados brancos 10 x 10 pixels em cada canto das imagens;
- Incluir 1 quadrado preto 15X15 no centro das imagens

Importando Bibliotecas

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

Incorporando o negativo das imagens

```
In [ ]: # Open image
imageLena = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/Lena.jpg')
imageHouse = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/House.jpg')
imageCamera = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital de Imagem/Camera.jpg')

# convert image to numpy array
npImageLena = np.array(imageLena)
npImageHouse = np.array(imageHouse)
npImageCamera = np.array(imageCamera)

# Create negative image
npImageNegativeLena = np.array(npImageLena)
npImageNegativeLena = 255 - npImageNegativeLena;
npImageNegativeHouse = np.array(npImageHouse)
```

```
npImageNegativeHouse = 255 - npImageNegativeHouse;
npImageNegativeCamera = np.array(npImageCamera)
npImageNegativeCamera = 255 - npImageNegativeCamera;
# Display images and their negatives using Matplotlib
fig, axs = plt.subplots(3, 2, figsize=(10, 15))

# Display original images
axs[0, 0].imshow(npImageLena, cmap='gray')
axs[0, 0].set_title('Original Lena')
axs[0, 0].axis('off')

axs[1, 0].imshow(npImageHouse, cmap='gray')
axs[1, 0].set_title('Original House')
axs[1, 0].axis('off')

axs[2, 0].imshow(npImageCamera, cmap='gray')
axs[2, 0].set_title('Original Camera')
axs[2, 0].axis('off')

# Display negative images
axs[0, 1].imshow(npImageNegativeLena, cmap='gray')
axs[0, 1].set_title('Negative Lena')
axs[0, 1].axis('off')

axs[1, 1].imshow(npImageNegativeHouse, cmap='gray')
axs[1, 1].set_title('Negative House')
axs[1, 1].axis('off')

axs[2, 1].imshow(npImageNegativeCamera, cmap='gray')
axs[2, 1].set_title('Negative Camera')
axs[2, 1].axis('off')

plt.tight_layout()
plt.show()
```

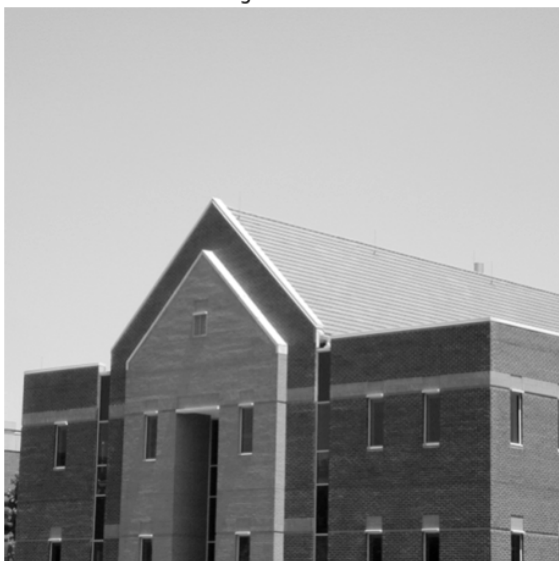
Original Lena



Negative Lena



Original House



Negative House



Original Camera



Negative Camera



Diminuir pela metade a intensidade dos pixels;

```
In [ ]: # divide by 2 pixels  
npImageLena = (npImageLena / 2).astype(int);
```

```

npImageHouse = (npImageHouse / 2).astype(int);
npImageCamera = (npImageCamera / 2).astype(int);

print(npImageLena.shape)
print(npImageHouse.shape)
print(npImageCamera.shape)

```

```
(300, 300)
```

```
(600, 600)
```

```
(512, 512)
```

Incluir 4 quadrados brancos 10 x 10 pixels em cada canto das imagens

```

In [ ]: imageLena = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital d
imageHouse = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital
imageCamera = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital

# convert image to numpy array
npImageLena = np.array(imageLena)
npImageHouse = np.array(imageHouse)
npImageCamera = np.array(imageCamera)
# Add white squares to the corners of the Camera image
npImageCamera[0:11,0:11] = 255
npImageCamera[501:512, 501:512] = 255
npImageCamera[0:11, 501:512] = 255
npImageCamera[501:512, 0:11] = 255

# Add white squares to the corners of the Lena image
npImageLena[0:11,0:11] = 255
npImageLena[589:600, 589:600] = 255
npImageLena[0:11, 589:600] = 255
npImageLena[589:600, 0:11] = 255

# Add white squares to the corners of the House image
npImageHouse[0:11,0:11] = 255
npImageHouse[589:600, 589:600] = 255
npImageHouse[0:11, 589:600] = 255
npImageHouse[589:600, 0:11] = 255

# Display the images using Matplotlib
fig, axs = plt.subplots(1, 3, figsize=(15, 5))

# Display Camera image with white squares
axs[0].imshow(npImageCamera, cmap='gray')
axs[0].set_title('Camera with Squares')
axs[0].axis('off')

# Display Lena image with white squares
axs[1].imshow(npImageLena, cmap='gray')
axs[1].set_title('Lena with Squares')
axs[1].axis('off')

# Display House image with white squares
axs[2].imshow(npImageHouse, cmap='gray')
axs[2].set_title('House with Squares')
axs[2].axis('off')

```

```
plt.tight_layout()
plt.show()
```

Camera with Squares



Lena with Squares



House with Squares



Incluir 1 quadrado preto 15X15 no centro das imagens

```
In [ ]: imageLena = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital d
imageHouse = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital
imageCamera = Image.open('/Meu Drive/Faculdade/Aula/2023.2/Processamento Digital

# convert image to numpy array
npImageLena = np.array(imageLena)
npImageHouse = np.array(imageHouse)
npImageCamera = np.array(imageCamera)

npImageLena = (npImageLena / 2).astype(int);
npImageHouse = (npImageHouse / 2).astype(int);
npImageCamera = (npImageCamera / 2).astype(int);

npImageLena[142:157,142:157] = 0
npImageCamera[248:263,248:263] = 0
npImageHouse[292:307,292:307] = 0

# Display the images using Matplotlib
fig, axs = plt.subplots(1, 3, figsize=(15, 5))

# Display Camera image with white squares
axs[0].imshow(npImageCamera, cmap='gray')
axs[0].set_title('Camera with Squares black')
axs[0].axis('off')

# Display Lena image with white squares
axs[1].imshow(npImageLena, cmap='gray')
axs[1].set_title('Lena with Squares black')
axs[1].axis('off')

# Display House image with white squares
axs[2].imshow(npImageHouse, cmap='gray')
axs[2].set_title('House with Squares black')
axs[2].axis('off')

plt.tight_layout()
plt.show()
```

Camera with Squares black



Lena with Squares black



House with Squares black

