INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO, CÂMPUS BIRIGUI - SP
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

ISADORA DISPOSTI BUENO DOS SANTOS

EXERCÍCIOS - FUNDAMENTOS IMAGENS

BIRIGUI - SP
2023

Link GitHub:

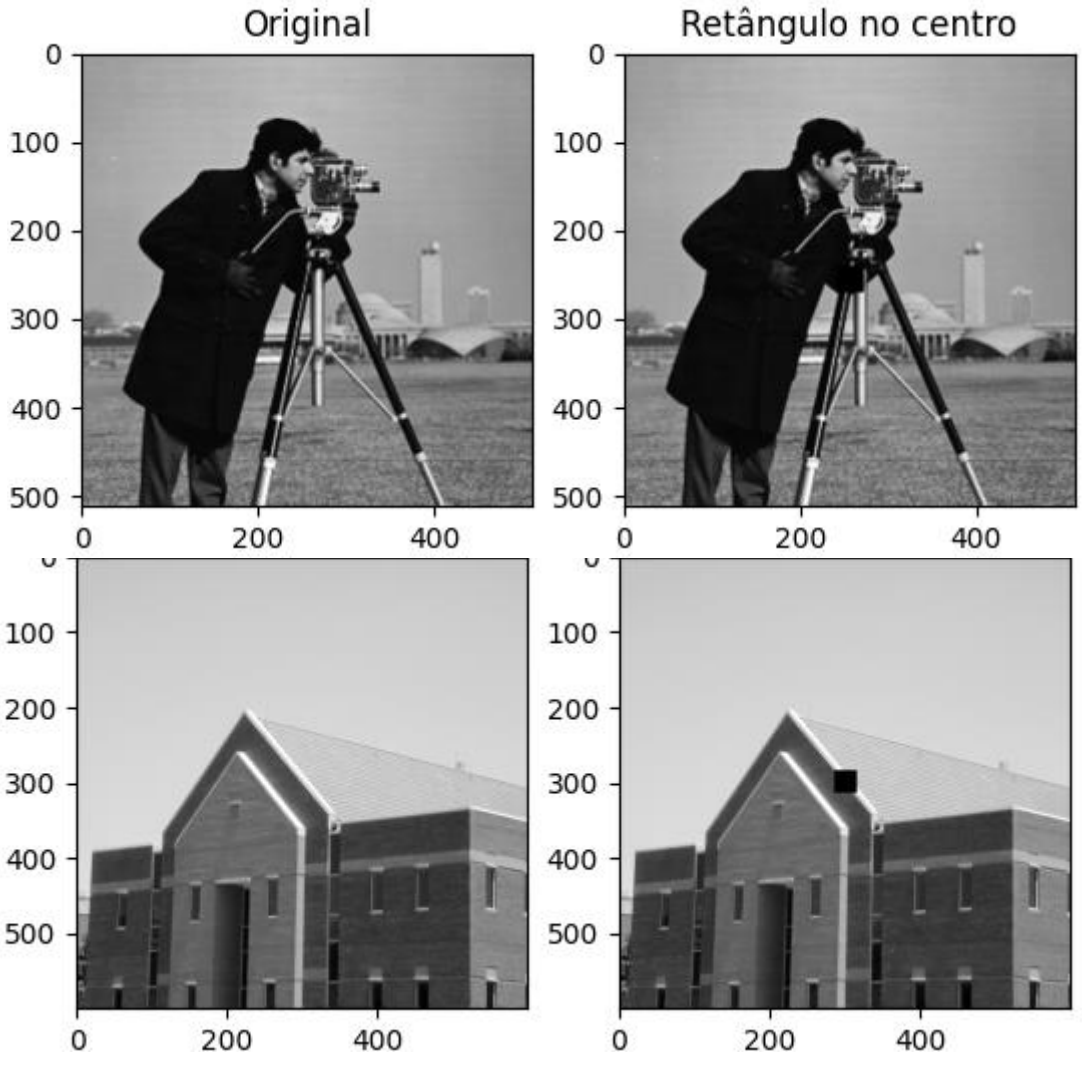
https://github.com/isadoradisposti/Fundamentos_De_Imagem.git

1. **[OPERAÇÃO PONTO A PONTO]:**
    1. Calcular o negativo das imagens;

    2. Diminuir pela metade a intensidade dos pixels;

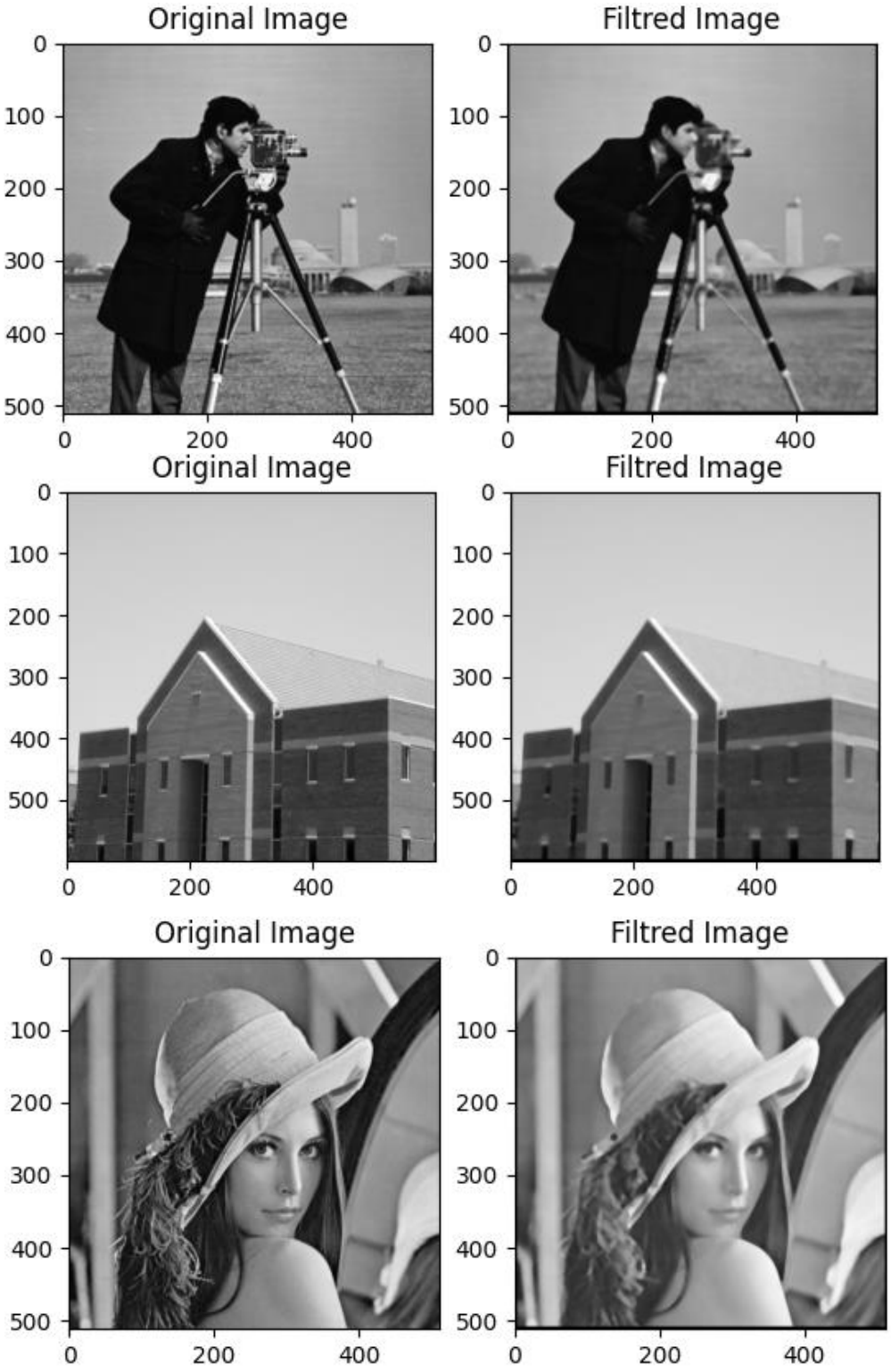
3. Incluir 4 quadrados brancos 10 x 10 pixels em cada canto das imagens;

4. Incluir 1 quadrado preto 15X15 no centro das imagens.

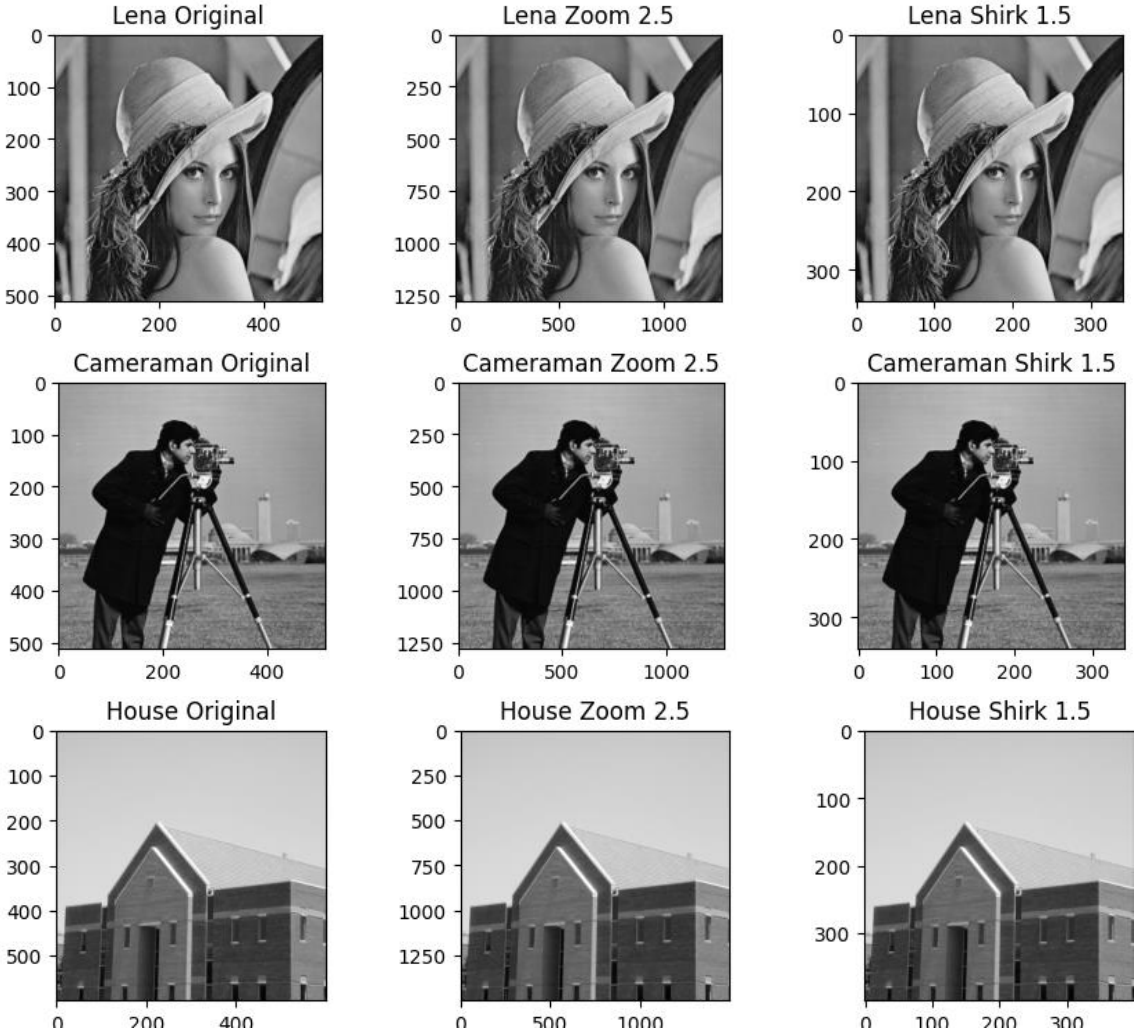


2. **[OPERAÇÃO POR VIZINHANÇA]:** Utilizar kernel 3x3 pixels e desconsiderar pixels das extremidades. Para cada filtro implementar utilizando apenas numpy, utilizando pillow, utilizando opencv e utilizando scipy.
    1. Calcular o filtro da média;

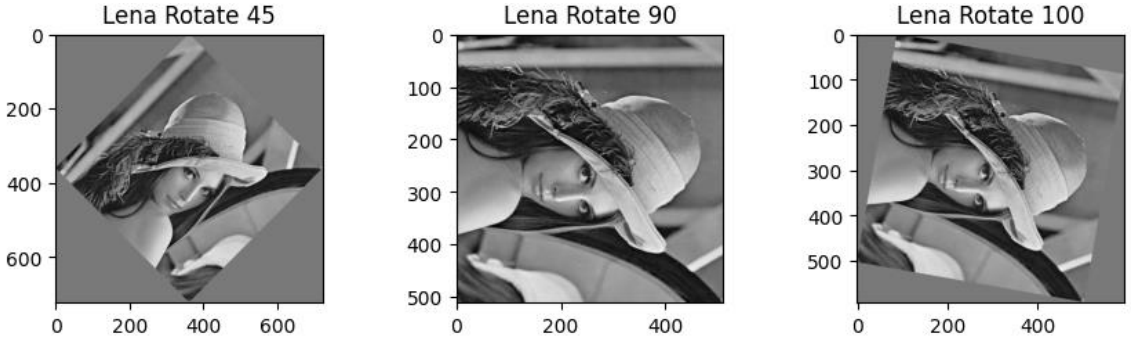
2. Calcular o filtro da mediana;

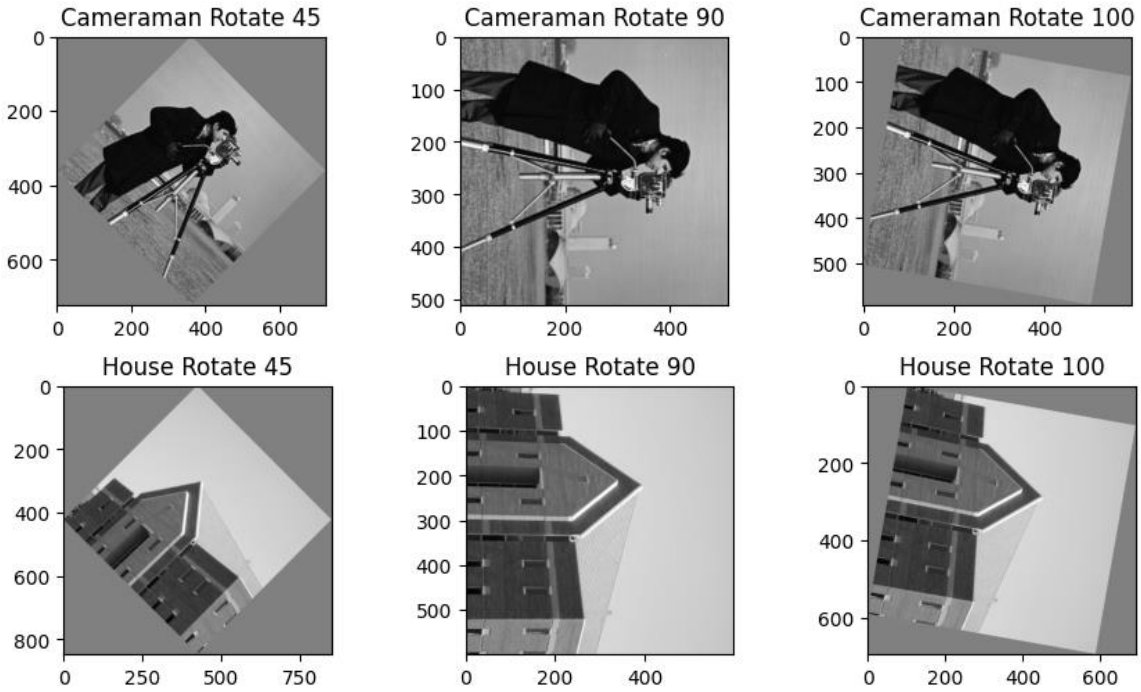


3. **[TRANSFORMAÇÕES GEOMÉTRICAS]:** Para cada filtro implementar utilizando apenas numpy, utilizando pillow, utilizando opencv e utilizando scipy.
    1. Escala: Redução em 1.5x e aumentar em 2.5x;

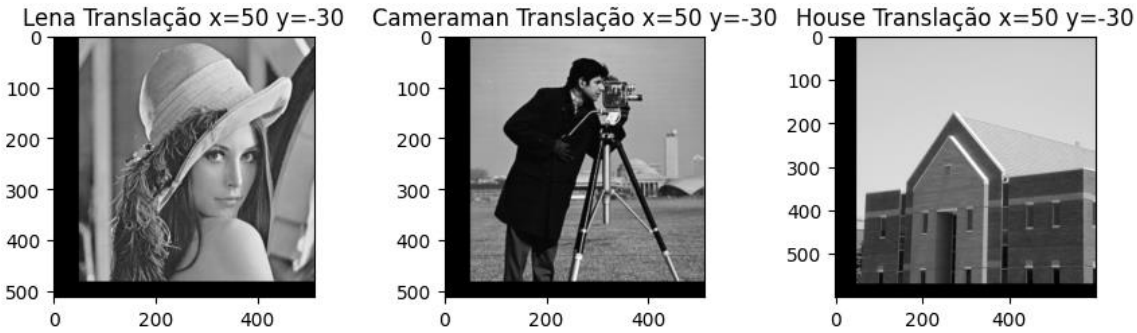


    2. Rotação em 45º, 90º e 100º;

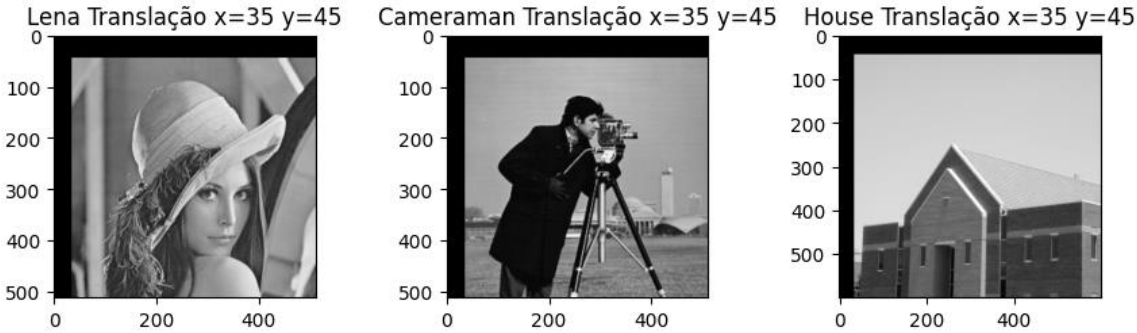




3. Translação utilizar os parâmetros que quiser nas coordenadas x e y;



4. Translação em 35 pixel no eixo X, 45 eixo Y;



Códigos VS Code:

[OPERAÇÃO PONTO A PONTO]:

1)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from scipy import ndimage

def main():
    #variáveis recebem o endereço das imagens
    lena =
np.array(Image.open('image/lena_gray_512.tif'))
    cam =
np.array(Image.open('image/cameraman.tif'))
    house = np.array(Image.open('image/house.tif'))

    #Transforma a foto da lena em negativa
    npLenaNegative = lena
    npLenaNegative = 255 - npLenaNegative

    #Transforma a foto do cameraman em negativa
    npCamNegative = cam
    npCamNegative = 255 - npCamNegative

    #Transforma a foto da house em negativa
    npHouseNegative = house
    npHouseNegative = 255 - npHouseNegative

    #Preparando para exibir as imagens pelo
matplotlib
    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt1.title.set_text("Lena Negativa")
```

```
    plt2 = plt.subplot(1, 3, 2)
    plt2.title.set_text("Cameraman Negativa")
    plt3 = plt.subplot(1, 3, 3)
    plt3.title.set_text("House Negativa")
    plt.subplots_adjust(wspace=0.5)

    plt1.imshow(npLenaNegative, cmap="gray")
    plt2.imshow(npCamNegative, cmap="gray")
    plt3.imshow(npHouseNegative, cmap="gray")

    plt.show()

if __name__ == "__main__":
    main()
```

2)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from scipy import ndimage

def main():
    #variáveis recebem o endereço das imagens
    lena =
np.array(Image.open('image/lena_gray_512.tif'))
    cam =
np.array(Image.open('image/cameraman.tif'))
    house = np.array(Image.open('image/house.tif'))

    #Intensidade da idade cortada na metade
    npLenaPixelImage = (lena / 2).astype(int)
    npCamPixelImage = (cam / 2).astype(int)
    npHousePixelImage = (house / 2).astype(int)
```

```
    #Exibir as figuras
    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt2 = plt.subplot(1, 3, 2)
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("Lena Com Intensidade
Reduzida")
    plt2.title.set_text("Cameraman Com Intensidade
Reduzida")
    plt3.title.set_text("House Com Intensidade
Reduzida")
    plt.subplots_adjust(wspace=0.5)

    plt1.imshow(npLenaPixelImage, cmap="gray")
    plt2.imshow(npCamPixelImage, cmap="gray")
    plt3.imshow(npHousePixelImage, cmap="gray")

    plt.show()

if __name__ == "__main__":
    main()
```

3)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from scipy import ndimage

def main():
    #variaveis recebem o endereço das imagens
    lena =
np.array(Image.open('image/lena_gray_512.tif'))
```

```
    cam =
np.array(Image.open('image/cameraman.tif'))
    house = np.array(Image.open('image/house.tif'))

    #Inserindo os quadrados nos cantos na imagem da
lena
    sizeLena = lena.shape[0]
    lenaCQuadrado = lena.copy()
    lenaCQuadrado[0:10, 0:10] = 255
    lenaCQuadrado[0:10, sizeLena-10:sizeLena] = 255
    lenaCQuadrado[sizeLena-10:sizeLena, 0:10] = 255
    lenaCQuadrado[sizeLena-10:sizeLena,sizeLena-
10:sizeLena] = 255
    plt.imshow(lenaCQuadrado, cmap="gray")
    plt.show()

    #Inserindo os quadrados nos cantos na imagem do
cameraman
    sizeCam = cam.shape[0]
    CamCQuadrado = cam.copy()
    CamCQuadrado[0:10, 0:10] = CamCQuadrado[0:10,
sizeCam-10:sizeCam]
    CamCQuadrado[sizeCam-10:sizeCam, 0:10]
    CamCQuadrado[sizeCam-10:sizeCam,sizeCam-
10:sizeCam] = 255
    plt.imshow(CamCQuadrado, cmap="gray")
    plt.show()

    #Inserindo os quadrados nos cantos na imagem da
house
    sizeHouse = house.shape[0]
    HouseCQuadrado = house.copy()
    HouseCQuadrado[0:10, 0:10] =
HouseCQuadrado[0:10,
```

```
    sizeHouse-10:sizeHouse] = 255
    HouseCQuadrado[sizeHouse-10:sizeHouse, 0:10]
    HouseCQuadrado[sizeHouse-
10:sizeHouse,sizeHouse-10:sizeHouse] = 255
    plt.imshow(HouseCQuadrado, cmap="gray")

    plt.show()

if __name__ == "__main__":
    main()
```

4)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt


def print_image(original, rect):
    plt.subplot(1, 2, 1)
    plt.title("Original")
    plt.imshow(original, cmap="gray")
    plt.subplot(1, 2, 2)
    plt.title("Retângulo no centro")
    plt.imshow(rect, cmap="gray")
    plt.show()


def rect_on_center(array):
    shape = array.shape
    shape = (shape[0] // 2, shape[1] // 2)
    array[shape[0] - 15 : shape[0] + 15, shape[1] -
15 : shape[1] + 15] = 0
    return array
```

```
def main():
    path = "C:/Users/isado/OneDrive/Área de
Trabalho/Projeto imagem/image"

    # Parâmetros
    lena = Image.open(path + "/lena_gray_512.tif")
    house = Image.open(path + "/house.tif")
    cameraman = Image.open(path + "/cameraman.tif")

    # Conversão para array
    lena_array = np.array(lena)
    house_array = np.array(house)
    cameraman_array = np.array(cameraman)

    # retângulo branco no centro
    lena_intensity =
rect_on_center(lena_array.copy())
    donout_intensity =
rect_on_center(house_array.copy())
    cameraman_intensity =
rect_on_center(cameraman_array.copy())

    # Exibição com matplotlib - original e com
retângulo preto no centro
    # Lena
    plt.figure("Lena")
    print_image(lena_array, lena_intensity)

    # house
    plt.figure("house")
    print_image(house_array, donout_intensity)

    # Cameraman
    plt.figure("Cameraman")
```

```
    print_image(cameraman_array,
cameraman_intensity)


if __name__ == "__main__":
    main()
```

[OPERAÇÃO POR VIZINHANÇA]:

1)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

def main():
    image_pillow =
Image.open('image/lena_gray_512.tif')
    f_image_nd = np.array(image_pillow)
    g_image_nd = np.zeros(f_image_nd.shape)

    #neighborhood operation (operação por
vizinhança)
    l = f_image_nd.shape[0]
    c = f_image_nd.shape[1]
    k = 3
    #print("Imagem")

    #print(f_image_nd[0:5,0:5])
    for x in range(k, l-k): #linhas
        for y in range(k, c-k): #colunas
            s_xy = f_image_nd[x-k:x+k+1,y-k:y+k+1]
            g_image_nd[x,y] =
np.mean(s_xy).astype(int)
            #print('janela')
            #print(s_xy)
```

```
    #create two columns plot
    fig = plt.figure()
    plt1 = plt.subplot(1,2,1)
    plt2 = plt.subplot(1,2,2)
    plt1.title.set_text('Original Image')
    plt2.title.set_text('Filtred Image')

    plt1.imshow(f_image_nd, cmap='gray')
    plt2.imshow(g_image_nd, cmap='gray')
    plt.show()

if __name__ == "__main__":
    main()
```

2)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

def main():
    image_pillow =
Image.open('image/lena_gray_512.tif')
    f_image_nd = np.array(image_pillow)
    g_image_nd = np.zeros(f_image_nd.shape)

    #neighborhood operation (operação por
vizinhança)
    l = f_image_nd.shape[0]
    c = f_image_nd.shape[1]
    k = 3
    #print("Imagem")

    #print(f_image_nd[0:5,0:5])
```

```
    for x in range(k, l-k): #linhas
        for y in range(k, c-k): #colunas
            s_xy = f_image_nd[x-k:x+k+1,y-k:y+k+1]
            g_image_nd[x,y] =
np.median(s_xy).astype(int)
            #print('janela')
            #print(s_xy)

    #create two columns plot
    fig = plt.figure()
    plt1 = plt.subplot(1,2,1)
    plt2 = plt.subplot(1,2,2)
    plt1.title.set_text('Original Image')
    plt2.title.set_text('Filtred Image')

    plt1.imshow(f_image_nd, cmap='gray')
    plt2.imshow(g_image_nd, cmap='gray')
    plt.show()

if __name__ == "__main__":
    main()
```

[TRANSFORMAÇÕES GEOMÉTRICAS]

1)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from scipy import ndimage

def main():
    #variaveis recebem o endereço das imagens
    lena =
np.array(Image.open('image/lena_gray_512.tif'))
```

```
    cam =
np.array(Image.open('image/cameraman.tif'))
    house = np.array(Image.open('image/house.tif'))

    #Mudança de zoom lena 1,5 e 2,5
lena_change_sizes
    lena_change_sizes = lena.copy()
    lena_zoom = ndimage.zoom(lena_change_sizes,
(2.5, 2.5))
    lena_shrink = ndimage.zoom(lena_change_sizes,
(1/1.5, 1/1.5))
    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt2 = plt.subplot(1, 3, 2)
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("Lena Original")
    plt2.title.set_text("Lena com Zoom 2.5")
    19
    plt3.title.set_text("Lena com Shirk 1.5")
    plt.subplots_adjust(wspace=0.5)

    plt1.imshow(lena, cmap="gray")
    plt2.imshow(lena_zoom, cmap="gray")
    plt3.imshow(lena_shrink, cmap="gray")

    cameraman_change_size = cam.copy()
    cameraman_zoom =
ndimage.zoom(cameraman_change_size, (2.5, 2.5))
    cameraman_shrink =
ndimage.zoom(cameraman_change_size, (1/1.5,
    1/1.5))

    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
```

```
    plt2 = plt.subplot(1, 3, 2)
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("Cameraman com Original")
    plt2.title.set_text("Cameraman com Zoom 2.5")
    plt3.title.set_text("Cameraman com Shirk 1.5")
    plt.subplots_adjust(wspace=0.5)

    plt1.imshow(cam, cmap="gray")
    plt2.imshow(cameraman_zoom, cmap="gray")
    plt3.imshow(cameraman_shrink, cmap="gray")

    house_chance_sizes = house.copy()
    house_zoom = ndimage.zoom(house_chance_sizes,
(2.5, 2.5))
    house_shrink = ndimage.zoom(house_chance_sizes,
(1/1.5, 1/1.5))
    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt2 = plt.subplot(1, 3, 2)
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("House Original")
    plt2.title.set_text("House com Zoom 2.5")
    plt3.title.set_text("House com Shirk 1.5")

    plt.subplots_adjust(wspace=0.5)
    plt1.imshow(house, cmap="gray")
    plt2.imshow(house_zoom, cmap="gray")
    plt3.imshow(house_shrink, cmap="gray")

    plt.show()

if __name__ == "__main__":
    main()
```

2)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from scipy import ndimage

def main():
    #variaveis recebem o endereço das imagens
    lena =
np.array(Image.open('image/lena_gray_512.tif'))
    cam =
np.array(Image.open('image/cameraman.tif'))
    house = np.array(Image.open('image/house.tif'))

    #Inclinação da foto lena
    lena_rotacao = lena.copy()
    lena_np_rotacao_45 =
ndimage.rotate(lena_rotacao, -45, cval=128)
    lena_np_rotacao_90 =
ndimage.rotate(lena_rotacao, -90, cval=128)
    lena_np_rotacao_100 =
ndimage.rotate(lena_rotacao, -100, cval=128)

    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt2 = plt.subplot(1, 3, 2)
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("Lena rotação 45")
    plt2.title.set_text("Lena rotação 90")
    plt3.title.set_text("Lena rotação 100")

    plt1.imshow(lena_np_rotacao_45, cmap="gray")
    plt2.imshow(lena_np_rotacao_90, cmap="gray")
    plt3.imshow(lena_np_rotacao_100, cmap="gray")
```

```
    plt.subplots_adjust(wspace=0.5)

    #Inclinação da foto cameraman
    cameraman_rotacao = cam.copy()
    cameraman_np_rotacao_45 =
ndimage.rotate(cameraman_rotacao, -45,
    cval=128)
    cameraman_np_rotacao_90 =
ndimage.rotate(cameraman_rotacao, -90,
    cval=128)
    cameraman_np_rotacao_100 =
ndimage.rotate(cameraman_rotacao, -100,
    cval=128)

    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt2 = plt.subplot(1, 3, 2)
    22
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("Cameraman rotação 45")
    plt2.title.set_text("Cameraman rotação 90")
    plt3.title.set_text("Cameraman rotação 100")
    plt1.imshow(cameraman_np_rotacao_45,
cmap="gray")
    plt2.imshow(cameraman_np_rotacao_90,
cmap="gray")
    plt3.imshow(cameraman_np_rotacao_100,
cmap="gray")
    plt.subplots_adjust(wspace=0.5)

    #Inclinação da foto house
    house_rotacao = house.copy()
    house_np_rotacao_45 =
ndimage.rotate(house_rotacao, -45, cval=128)
```

```
    house_np_rotacao_90 =
ndimage.rotate(house_rotacao, -90, cval=128)
    house_np_rotacao_100 =
ndimage.rotate(house_rotacao, -100,
    cval=128)

    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt2 = plt.subplot(1, 3, 2)
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("House rotação 45")
    plt2.title.set_text("House rotação 90")
    plt3.title.set_text("House rotação 100")
    plt.subplots_adjust(wspace=0.5)

    plt1.imshow(house_np_rotacao_45, cmap="gray")
    plt2.imshow(house_np_rotacao_90, cmap="gray")
    plt3.imshow(house_np_rotacao_100, cmap="gray")

    plt.show()

if __name__ == "__main__":
    main()
```

3)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from scipy import ndimage
def main():

    #variaveis recebem o endereço das imagens
```

```
    lena =
np.array(Image.open('image/lena_gray_512.tif'))
    cam =
np.array(Image.open('image/cameraman.tif'))
    house = np.array(Image.open('image/house.tif'))

    #Mover as coordenadas x e y para 200 e -45
    translacao_lena = lena.copy()
    translacao_cameraman = cam.copy()
    translacao_house = house.copy()
    shift_x = 200
    shift_y = -45
    shift_vector = [shift_y, shift_x]

    imagem_trasicionada_lena =
ndimage.shift(translacao_lena,
    shift_vector, mode='constant', cval=0)
    imagem_trasicionada_cameraman =
ndimage.shift(translacao_cameraman
    , shift_vector, mode='constant', cval=0)
    imagem_trasicionada_house =
ndimage.shift(translacao_house,
    shift_vector, mode='constant', cval=0)

    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt2 = plt.subplot(1, 3, 2)
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("Lena Translação x=200 y=-
45")
    plt2.title.set_text("Cameraman Translação x=200
y=-45")
    plt3.title.set_text("House Translação x=200 y=-
45")
```

```
    plt.subplots_adjust(wspace=0.5)

    plt1.imshow(imagem_trasicionada_lena,
cmap="gray")
    plt2.imshow(imagem_trasicionada_cameraman,
cmap="gray")
    plt3.imshow(imagem_trasicionada_house,
cmap="gray")

    plt.show()

if __name__ == "__main__":
    main()
```

4)

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from scipy import ndimage
def main():
    #variaveis recebem o endereço das imagens
    lena =
np.array(Image.open('image/lena_gray_512.tif'))
    cam =
np.array(Image.open('image/cameraman.tif'))
    house = np.array(Image.open('image/house.tif'))

    translacao_lena = lena.copy()
    translacao_cameraman = cam.copy()
    translacao_house = house.copy()
    shift_x = 35
    shift_y = 45
    shift_vector = [shift_y, shift_x]
```

```
    translated_image_lena =
ndimage.shift(translacao_lena,
    shift_vector, mode='constant', cval=0)
    translated_image_cam =
ndimage.shift(translacao_cameraman ,
    shift_vector, mode='constant', cval=0)
    translated_image_house =
ndimage.shift(translacao_house,
    shift_vector, mode='constant', cval=0)

    fig = plt.figure(figsize=(10, 5))
    plt1 = plt.subplot(1, 3, 1)
    plt2 = plt.subplot(1, 3, 2)
    plt3 = plt.subplot(1, 3, 3)
    plt1.title.set_text("Lena Translação x=35
y=45")
    plt2.title.set_text("Cameraman Translação x=35
y=45")
    plt3.title.set_text("House Translação x=35
y=45")
    plt.subplots_adjust(wspace=0.5)
    plt1.imshow(translated_image_lena, cmap="gray")
    plt2.imshow(translated_image_cam, cmap="gray")
    plt3.imshow(translated_image_house,
cmap="gray")

    plt.show()

if __name__ == "__main__":
    main()
```