Imagens Biomédicas – Lab02-Pyton

Tutores: André Arruda / Maíra Suzuka Kudo / Eric Rocha Santos

Professor: Matheus Cardoso Moraes

HISTOGRAMA, BRILHO E CONTRASTE

1 EXERCÍCIOS:

Criar e colocar na Pasta Aula02 as imagens Mamography.pgm e Stent.pgm.

1. **Dê um nome e salve o arquivo do Editor na pasta Aula02. Pode chamar: transformacao_intensidade.py**

   .....Aula02\transformacao_intensidade.py

   → No Editor(transformacao_intensidade.py), importe as bibliotecas abaixo:

   ```
   import numpy as np
   import matplotlib.pyplot as plt
   import cv2 # OpenCV
   import skimage
   import skimage.exposure
   ```

2. **→ No Editor, faça a leitura dessas imagens, atribuindo seus valores a duas matrizes *Mamo e Stent*.**
   Ex.:

   ```
   Mamo = cv2.imread('Mamography.pgm', 0)
   Stent = cv2.imread('Stent.pgm', 0)
   ```

   Obs.: indicar corretamente o caminho de diretórios onde estão contidas as imagens.

3. → No Editor, faça a normalização dessas imagens, atribuindo seus valores a duas matrizes *MamoN e StentN*.
   Ex.:

```
MamoN = skimage.img_as_float(Mamo)
StentN = skimage.img_as_float(Stent)
```

4. → No Editor, use as funções abaixo para exibir a imagem Mamo.

```
plt.figure(0)
plt.ylabel('linhas - M')
plt.xlabel('colunas - N')
plt.title('MamoN')
plt.imshow(MamoN, cmap='gray') # cmap='jet'
plt.colorbar()
```

5. → (Loop) Fazer uma combinação do condicional "*for*" que passe por cada pixel da imagem MamoN. Usar este loop para obter o Negativo da Imagem Mamo. Para tanto faça:

   a. Use a função "np.shape" para obter o número de linhas M e colunas N da imagem Mamo.

   b. Use a função "np.zeros" para Criar Matriz para Negativo da Mamo
      NegativoMamoPorPixel = np.zeros((M,N), float)
      NegativoMamoDireto = np.zeros((M,N), float)

   c. Dentro da combinação de "*for", varra todas as linas l e colunas c, e* crie uma operação que obtenha o negativo da imagem MamoN por pixel, e faça exibição da imagem.

```
for l in range(M):
    for c in range(N):
        NegativoMamoPorPixel [l,c] = equação?

plt.figure(1)
plt.ylabel('linhas - M')
plt.xlabel('colunas - N')
plt.title('NegativoMamoPorPixel')
plt.imshow(NegativoMamoPorPixel , cmap='gray') # cmap='jet'
plt.colorbar()
```

6. Crie uma operação que obtenha o negativo da imagem Mamo, de forma direta, sem precisar do loop de for, chame-a de NegativoMamoDireto, e faça uma nova exibição.

```
NegativoMamoDireto = equação?

plt.figure(2)
plt.ylabel('linhas - M')
plt.xlabel('colunas - N')
plt.title('NegativoMamoDireto')
plt.imshow(NegativoMamoDireto , cmap='gray') # cmap='jet'
plt.colorbar()
```

O resultado foi o mesmo? Qual é mais rápido?

7. → Crie uma função chamada fazerHistograma, dentro de uma biblioteca chamada bibFuncoesHistograma.py. Na qual a entrada de dados será uma das imagens, e a função retornará um vetor chamado histograma, com o histograma da imagem, ou seja a distribuição de valores de intensidade da imagem. As classes serão divididas pelos 256 níveis de intensidades.

   a. Criar Biblioteca → Abrir um arquivo dentro da pasta Aula02 e chamar de bibFuncoesHistograma.py

   b. Dentro desta biblioteca, criar a função: *fazerHistograma* usando *def* da seguinte forma:

```
def fazerHistograma(imagem, M, N):

    # importar bibliotecas
    import numpy as np
    import matplotlib.pyplot as plt
    import cv2 # OpenCV

    # declarar vetor histograma inicialmente tudo com 0
    histograma = np.zeros((256), int)

    fazer lógica para criar histograma

    return histograma
```

   Obs.: Atenção a identação

   c. Programa Principal: chame a função fazerHistograma

```
# import lib
import bibFuncoesHistograma
# chame a função com as entradas e saídas
histograma = bibFuncoesHistograma.fazerHistograma(Mamo, M, N)
```

   Obs.: Para este exercício, mamo está em 8bits

d. **Exibir o vetor usando a função *stem***

```
# plota hstograma
plt.figure(3)
plt.stem(histograma, use_line_collection=True)
plt.show()
```

e. **Use a função em vermelho da biblioteca skiimage.exposure para comparar com seu método*, chame de histograma2***

```
histograma2 = skimage.exposure.histogram(Mamo)
x = histograma2[1] # Classes
y = histograma2[0] # Numero de Ocorrência
plt.figure(4)
plt.stem(x, y, use_line_collection=True)
plt.ylabel('Numero de Ocorrência')
plt.xlabel('Classes')
plt.show()
```

f. **Refaça o exercício anterior*, mas agora para as classes da imagem normalizada entre 0 e 1, chame de histograma3. Verifique se as classes ficaram entre 0 e 1.***

```
Histograma3 = skimage.exposure.histogram(MamoN)
x = histograma3[1] # Classes
y = histograma3[0] # Numero de Ocorrência
plt.figure(5)
plt.stem(x, y, use_line_collection=True)
plt.ylabel('Numero de Ocorrência')
plt.xlabel('Classes')
plt.show()
```

8. → Faça uma operação que aumente o nível de cinza(brilho) da imagem "StentN" em mais 0.2 níveis de intensidade, considerando ela entre 0 - 1.

StentBrilhoN = equação?

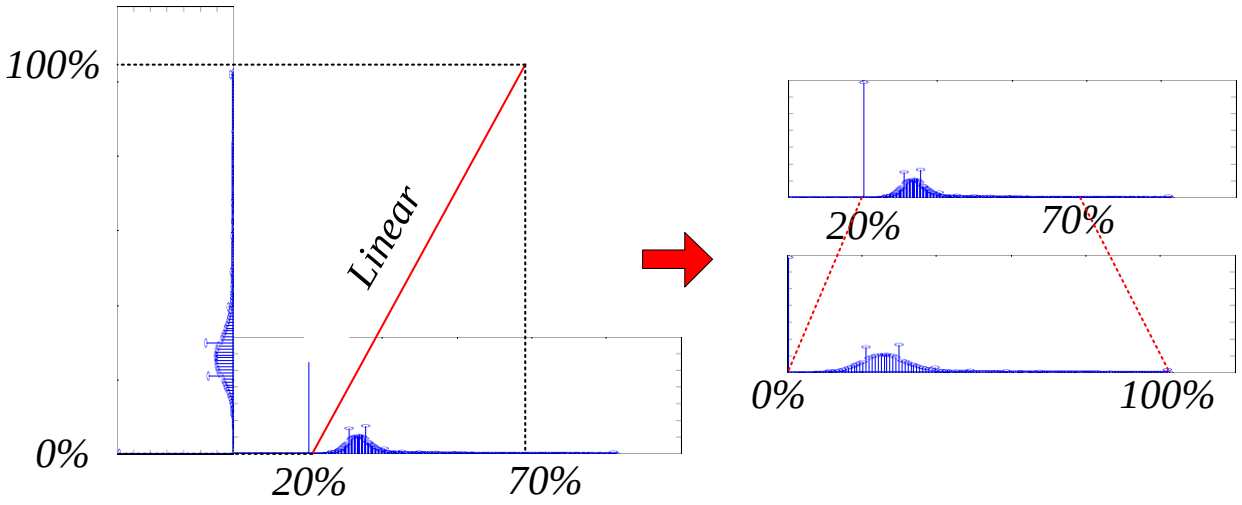
Renormalizar intensidade para não sair do range, incluir níveis de intensidades fora do rage

StentBrilhoN = skimage.exposure.rescale_intensity(StentBrilhoN, in_range=(0,1))

Plote a Imagem e o histograma desta imagem para verificar se houve aumento de brilho?

9. → Combine a funçãp abaixo para fazer um aumento de contraste da imagem StentBrilho. Usando a função "rescale_intensity" recorte os valores de intensidade entre 20% e 70%, da faixa de intensidade respectivamente, remanejando os valores intermediários de forma linear (reta- Figura abaixo) para ocupar 100% do histograma da saída.

# Ajuste de Contraste
StentAlongada = skimage.exposure.rescale_intensity(StentBrilhoN, in_range=(0.2,0.7))

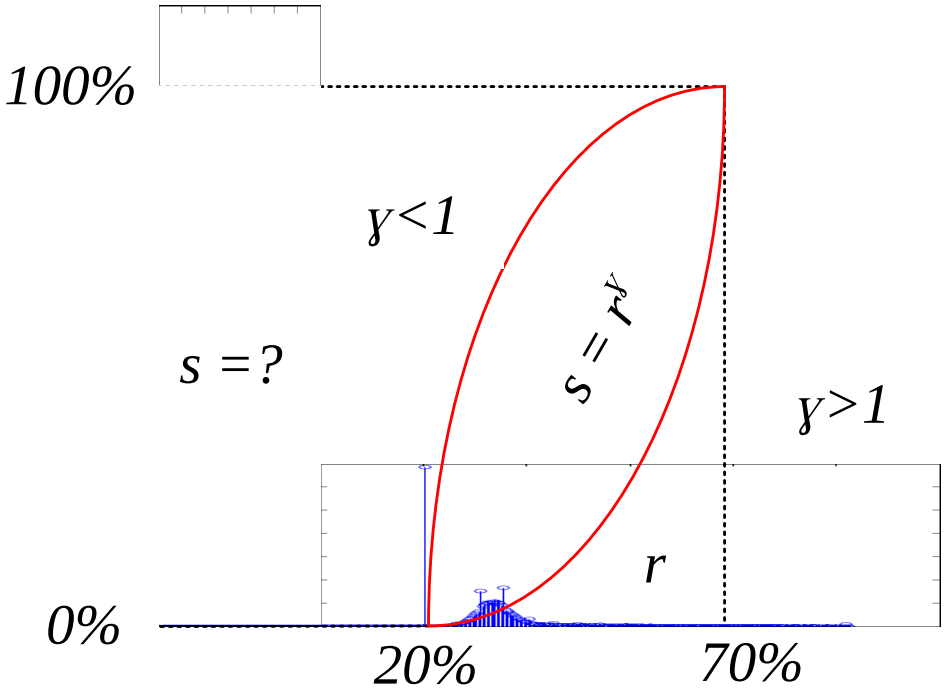


Plote a Imagem e o histograma desta imagem para verificar se houve aumento de contraste.

10.→Ainda usando a função "adjust_gamma", altere a variável gamma (Figura) para valores menores e maiores que 1, na imagem StentBrilho.

StentGamma = skimage.exposure.adjust_gamma(StentAlongada, 1)

Plote a Imagem e o histograma desta imagem para verificar se houve aumento de contraste? E em qual situação, o stent da imagem é mais destacado? Por quê?



11.→ Usando o intervalo de valores do histograma do exercício 8, faça um estudo para avaliar qual o valor de gamma que mais distingue o stent do resto da imagem sem degrada-lo. Para isto, ainda usando a função imadjust, altere a variável gamma de acordo com valores descritos abaixo (Tabela). Para cada valor de gamma faça uma inspeção visual (subjetiva), pontuando de 1 á 5 os dois atributos. Sendo nota 1 para quando não haver alteração perceptível, e 5 para quando a distinção e/ou degradação forem muito relevantes. Baseado nessas notas, qual valor de gamma seria a melhor opção? Explique.

Imagens com stent										
gamma										
	nível de distinção do stent					**nível de degradação do stent**				
	1	**2**	3	4	5	1	2	3	4	5
0,1										
0,2										
0,3										
0,4										
0,5										
0,6										
0,7										
0,8										
0,9										
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										