PRÁTICA 2.

1) Visualização de Histogramas.

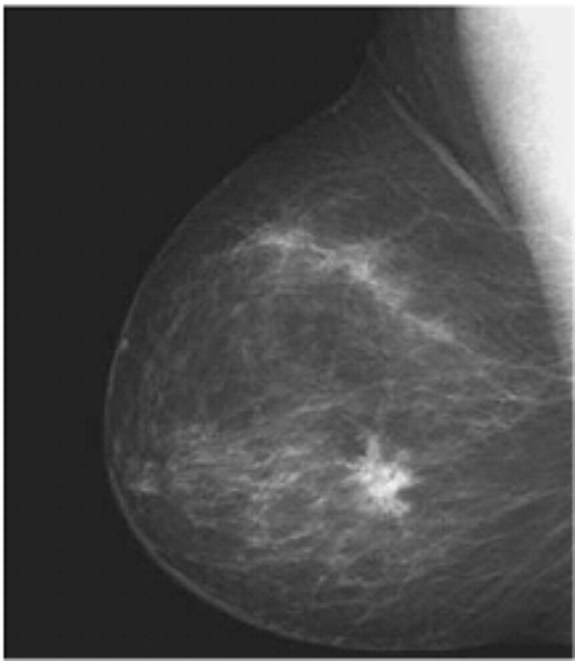

Figura 1 - mammogram.bmp

T_1: Ler a imagem mammogram.bmp e gerar seu histograma.

```
f = imread('mammogram.bmp');
imfinfo mammogram.bmp
imshow(f)
imhist(f)
```

E_1: Gerar o histograma anterior com 256 níveis de cinza em outros formatos:

a) Formato de barras com 10 píxels de largura (função ***bar***)
b) Formato de hastes a cada 10 píxels (função ***stem***)
c) Formato de uma curva contínua (função ***plot***)

E_2: Alterar os Histogramas gerados anteriormente:

a) Adicionando um título a cada histograma (função ***title***)
b) Adicionando rótulos para os eixos (funções ***xlabel*** e ***ylabel***)
c) Adicionando um texto na figura (função ***text***)
d) Expandindo a escala dos eixos horizontal e vertical (função ***axis***)

2) Transformação de Intensidades.

As técnicas de processamento no domínio espacial operam diretamente nos pixels da imagem. A expressão geral para a **Função de Transformação** nos níveis de cinza pode ser dada por:

$$g(x, y) = T[\, f\,(x, y)]$$

onde ***f(x,y)*** é a imagem de entrada e ***g(x,y)*** é a imagem de saída ou imagem processada.
T é um operador em ***f.***

A função do MatLab que realiza transformações de intensidade nos níveis de cinza de uma imagem é a ***imadjust*** que tem a seguinte sintaxe:

g = imadjust(f, [low_in high_in], [low_out high_out], gamma)

cuja função de transferência é vista na figura 1.

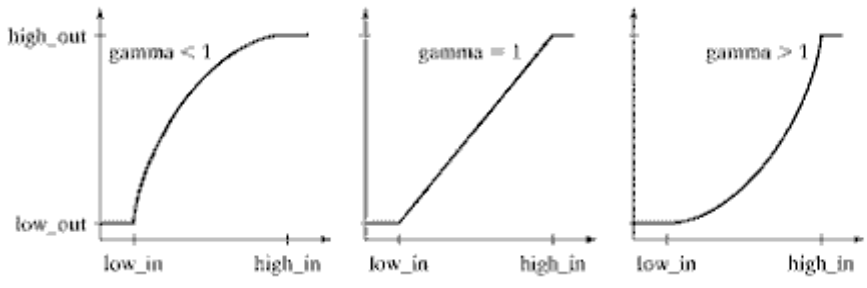

Figura 2 - Alguns mapeamentos disponíveis na função *imadjust*

E_3: Mostrar graficamente a Função de Transformação de Intensidades -T[f(x,y)] e a imagem gerada em cada um dos exemplos.

a) g1 = imadjust(f, [0 1], [1 0])
b) g2 = imadjust(f, [0.5 0.75], [0 1])
c) g3 = imadjust(f, [ ], [ ], 2)

3) Equalização do Histograma.

A equalização de histogramas no MatLab é implementada através da função:

$$g = histeq(f, nlev)$$

onde ***f*** é a imagem de entrada e ***nlev*** é o número de níveis de intensidades especificados para a imagem de saída.

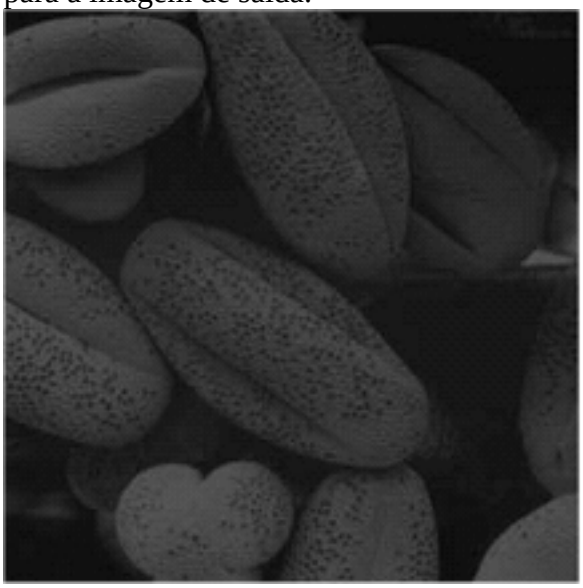

Figura 3 - polem.bmp

T_2: Equalizar a imagem polem.bmp dada na Figura 3.

```
g = imread('polem.bmp');
figure, imshow(g)
figure, imhist(g)
ylim('auto')
geq = histeq(g,256);
figure, imshow(geq)
figure, imhist(geq)
ylim('auto')
```

E_4: Fornecer para o Exemplo anterior (T_2) :

a) Os dois histogramas (equalizado e não)
b) As duas Imagens (equalizada e não)
c) Comentar sobre a função e a necessidade da equalização
d) Explicar o uso do comando ***ylin*** do MatLab
e) O que acontece se a imagem equalizada for equalizada novamente?

A Função de Transformação da Equalização de Histograma é a Soma Cumulativa dos valores do histograma normalizado. Esta Função de Transformação transforma uma estreita faixa de níveis de intensidade de entrada em uma escala completa de níveis de intensidade de saída.

E_5: Executar e explicar o código de MatLab abaixo tanto relativo ao resultado obtido como a função de cada linha de comando.

```
g = imread('polem.bmp');
figure, imshow(g)
figure,imhist(g)
ylim('auto')
hnorm = imhist(g)./numel(g);
cdf = cumsum(hnorm);
x = linspace(0, 1, 256);
figure,plot(x,cdf)
axis([0 1 0 1])
set(gca, 'xtick', 0:.2:1)
set(gca, 'ytick', 0:.2:1)
xlabel('Valores de Intensidade de Entrada', 'fontsize', 9)
ylabel('Valores de Intensidade de Saída', 'fontsize', 9)
text(0.18, 0.5, 'Função de Transfomação', 'fontsize', 9)
```

4) Subtração de Imagens.

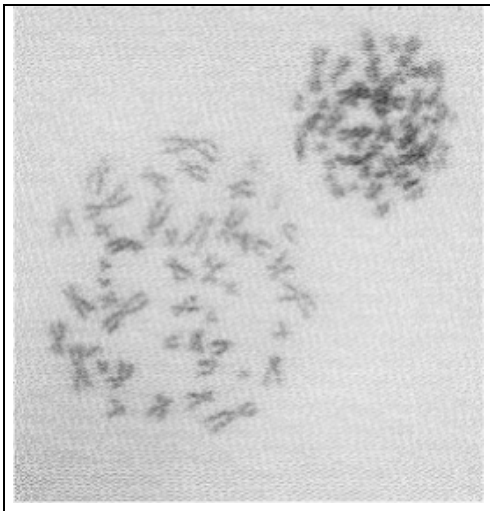

Figura 4 - Image2.bmp - Cromossomos

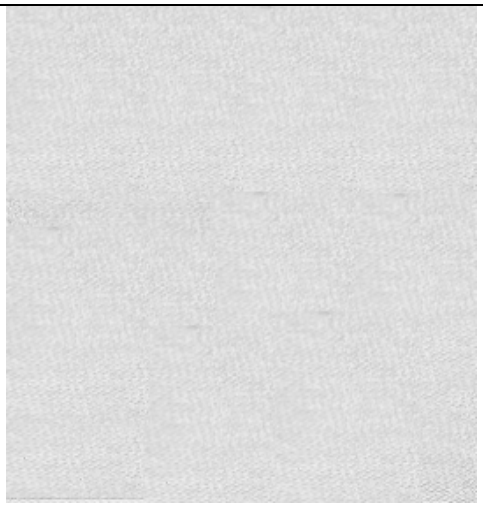

Figura 5 - Image3.bmp - Fundo da Imagem de Cromossomos

T_3: Subtrair a imagem do fundo.

```
g1 = imread('Image2.bmp')
g2 = imread('Image3.bmp')
g3 = imabsdiff(g2,g1);
g4 = imcomplement(g3);
figure('Name', 'Imagem Original'),imshow(g1)
figure('Name', 'Histograma da Imagem Original'), imhist(g1)
figure('Name','Imagem do Fundo'), imshow(g2)
figure('Name','Histograma da Imagem do Fundo'), imhist(g2)
figure('Name', 'Imagem Diferença'), imshow(g3)
figure('Name', 'Histograma da Imagem Diferença'), imhist(g3)
figure('Name','Imagem resultado da subtração complementada'), imshow(g4)
figure('Name','Histograma da Imagem complementada'), imhist(g4)
```

E_6: Para o Exemplo T_3 .

a) Fornecer e comentar cada imagem e seu histograma.
b) Mostrar o resultado da subtração usando ***g3=g2-g1 e g3=g1-g2***
c) Explicar o porque de se usar a função ***imabsdiff(g2,g1)*** e não apenas a diferença simples entre matrizes
d) Explicar o que faz a função ***g4 = imcomplement(g3)****;*

5) Transformação nos níveis de cinza – Visualização pelo Histograma.

E_7: Alterações Globais no Brilho.

d) Clarear a imagem polem.bmp de 128 níveis de cinza.
e) Equalizar as imagens (clareada e não) e gerar seus histogramas. Concluir a respeito.
f) Alterar o brilho para mais 200 níveis de cinza e refazer o experimento. Concluir.

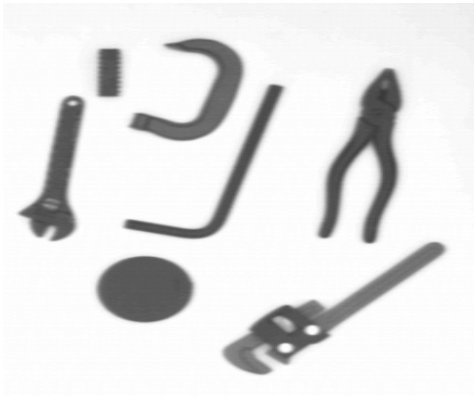

Figura 6 - ferramentas.bmp

E_8: Binarização (Threshold).

a) Binarizar a imagem ferramentas.bmp usando a função ***im2bw*** pelo Método do vale variando o limiar de 20% a 60% da escala de cinza. Mostrar os histogramas equivalentes.
b) Binarizar a imagem polem.bmp através da função ***graythresh*** (Método de Otsu). Mostrar o histograma equivalente.
c) Qual a diferença entre as duas metodologias?

6) Filtragem Espacial: Filtro Passa Baixa e Passa Alta.

Figura 7- Lena_ruido.bmp

O Toolbox de Processamento de Imagens do MATLAB implementa a Filtragem Espacial Linear através da função ***imfilter*** que possui a seguinte sintaxe:

g = imfilter(f,w, filtering_mode, boundary_options, size_options)

onde ***f*** é a imagem de entrada, ***w*** é a máscara do filtro e ***g*** é a imagem resultante. Os outros parâmetros são dados pela Tabela 1.

Options	Description
Filtering Mode	
`'corr'`	Filtering is done using correlation (see Figs. 3.13 and 3.14). This is the default.
`'conv'`	Filtering is done using convolution (see Figs. 3.13 and 3.14).
Boundary Options	
`P`	The boundaries of the input image are extended by padding with a value, `P` (written without quotes). This is the default, with value 0.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'circular'`	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
`'full'`	The output is of the same size as the extended (padded) image (see Figs. 3.13 and 3.14).
`'same'`	The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image (see Figs. 3.13 and 3.14). This is the default.

Tabela 1 - Opções da função *imfilter*

O filtro da Mediana pode ser implementado através da função ***medfilt2*** cuja sintaxe é:

g = medfilt2(f, [m n], padopt)

Onde ***[m n]*** define a vizinhança para a Mediana e ***padopt*** é a opção de borda da imagem (***padopt = 'zeros'*** → default, ***'symmetric'*** → a imagem é considerada simétrica em suas bordas, ***'indexed'*** → a borad é considerada 1 se f for da classe doublé e 0 caso contrário)

E_9: Suavização – Filtragem espacial passa baixa.
Para a imagem ruidosa da Figura 7

a) Filtrá-la através da Média de Vizinhança 5x5, 7x7, 9x9, 25x25, 31x31
b) Filtrá-la através da Mediana 5 x 5, 7x7
c) Filtrá-la através Média dos k-vizinhos de 25 (fazer k= 9,15,20)

- Concluir a respeito das 3 metodologias.

O Toolbox de Processamento de Imagens do MATLAB implementa também filtros 2-D pré-definidos através da função ***fspecial*** que gera uma máscara de filtro ***w*** através da seguinte sintaxe:

w = fspecial('type', parameters)

Onde ***'type'*** especifica o tipo do filtro e ***parameters*** são definidos pelo filtro especificado. A Tabela 2 mostra os filtros especiais que podem ser implementados.

Figura 8 - Moon.tif

Type	Syntax and Parameters
`'average'`	`fspecial('average', [r c])`. A rectangular averaging filter of size `r` × `c`. The default is 3 × 3. A single number instead of `[r c]` specifies a square filter.
`'disk'`	`fspecial('disk', r)`. A circular averaging filter (within a square of size `2r + 1`) with radius `r`. The default radius is 5.
`'gaussian'`	`fspecial('gaussian', [r c], sig)`. A Gaussian lowpass filter of size `r` × `c` and standard deviation `sig` (positive). The defaults are 3 × 3 and 0.5. A single number instead of `[r c]` specifies a square filter.
`'laplacian'`	`fspecial('laplacian', alpha)`. A 3 × 3 Laplacian filter whose shape is specified by `alpha`, a number in the range [0, 1]. The default value for `alpha` is 0.5.
`'log'`	`fspecial('log', [r c], sig)`. Laplacian of a Gaussian (LoG) filter of size `r` × `c` and standard deviation `sig` (positive). The defaults are 5 × 5 and 0.5. A single number instead of `[r c]` specifies a square filter.
`'motion'`	`fspecial('motion', len, theta)`. Outputs a filter that, when convolved with an image, approximates linear motion (of a camera with respect to the image) of `len` pixels. The direction of motion is `theta`, measured in degrees, counterclockwise from the horizontal. The defaults are 9 and 0, which represents a motion of 9 pixels in the horizontal direction.
`'prewitt'`	`fspecial('prewitt')`. Outputs a 3 × 3 Prewitt mask, `wv`, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: `wh = wv'`.
`'sobel'`	`fspecial('sobel')`. Outputs a 3 × 3 Sobel mask, `sv`, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: `sh = sv'`.
`'unsharp'`	`fspecial('unsharp', alpha)`. Outputs a 3 × 3 unsharp filter. Parameter `alpha` controls the shape; it must be greater than 0 and less than or equal to 1.0; the default is 0.2.

Tabela 2 - Filtros especiais da função *fspecial*

T_4: Laplaciano – Filtro Passa Alta (Image Enhancement).

Para a imagem da Figura 8, aplicar o filtro Laplaciano e um filtro passa alta(w8).

```
f = imread('Moon.tif');
w4 = fspecial('laplacian',0);
w8 = [1 1 1; 1 -8 1; 1 1 1];
f = im2double(f);
g4 = f - imfilter(f, w4, 'replicate');
g8 = f - imfilter(f, w8, 'replicate');
imshow(f)
figure, imshow(g4)
figure, imshow(g8)
```

E_10: Filtros Passa Alta.

a) Para o procedimento realizado em T_4, explicar a razão de se converter a classe da imagem para double.
b) Em T_4, o que faz o filtro w4 e w8? Qual a diferença entre eles?
c) Aplicar o detector de bordas de Sobel (horizontal, vertical e completo) nas imagens das Figuras 6, 7, 8 e 9 .
d) Aplicar o detector de Prewitt nas imagens das Figuras 6 , 7, 8 e 9 comparando-o com o detector de Sobel.

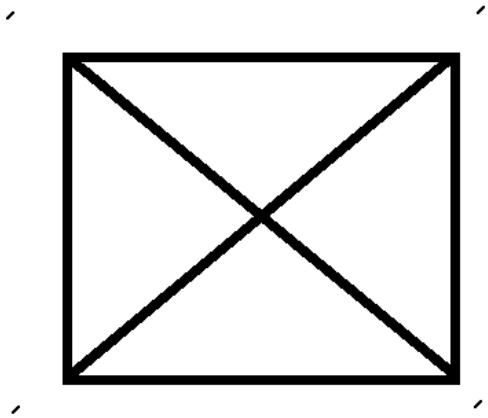

Figura 9 - teste.bmp

7) Média de Múltiplas Imagens

Uma imagem pode ser corrompida com ruído. A função do MATLAB que contamina uma imagem com ruído é a ***imnoise*** que tem a seguinte sintaxe:

fn = imnoise(f, type, parameters)

Onde ***f*** é a imagem original e ***type*** é o tipo de ruído conforme mostra a Tabela 3.

Value	Description
'gaussian'	Gaussian white noise
'localvar'	Zero-mean Gaussian white noise with an intensity-dependent variance
'poisson'	Poisson noise
'salt & pepper'	On and off pixels
'speckle'	Multiplicative noise

Tabela 3 - Tipos de ruídos

T_5: Média de Múltiplas Imagens

A partir da imagem original da Figura 6, contaminá-la com ruído 'salt & pepper' com 60% de probabilidade de ocorrência e fazer a média das imagens ruidosas.

```
k = imread('ferramentas.bmp');
f = im2double(k);
f1 = imnoise(f,'salt & pepper', 0.6);
f2 = imnoise(f,'salt & pepper', 0.6);
f3 = imnoise(f,'salt & pepper', 0.6);
f4 = imnoise(f,'salt & pepper', 0.6);
f5 = imnoise(f,'salt & pepper', 0.6);
f6 = imnoise(f,'salt & pepper', 0.6);
f7 = imnoise(f,'salt & pepper', 0.6);
f8 = imnoise(f,'salt & pepper', 0.6);
f9 = imnoise(f,'salt & pepper', 0.6);
f10 = imnoise(f,'salt & pepper', 0.6);
fm = (f1 + f2 + f3 + f4 + f5 + f6 + f7 + f8 + f9 + f10)/10;
subplot(2,3,1);imshow(f)
subplot(2,3,2);imshow(f1)
subplot(2,3,3);imshow(f2)
subplot(2,3,4);imshow(f3)
subplot(2,3,5);imshow(f4)
subplot(2,3,6);imshow(f5)
figure
subplot(2,3,1);imshow(f6)
subplot(2,3,2);imshow(f7)
subplot(2,3,3);imshow(f8)
subplot(2,3,4);imshow(f9)
subplot(2,3,5);imshow(f10)
subplot(2,3,6);imshow(fm)
```

E_11: Filtragem através da média de múltiplas imagens.

a) Para o treinamento T_5 alterar o valor da probabilidade de ocorrência do ruído entre 0 e 1 e concluir a respeito da média de múltiplas imagens.
b) Substituir o tipo do ruído por um ruído branco gaussiano com média zero e verificar o resultado concluindo a respeito da média de múltiplas imagens.