



Atividade #02

Vale nota, individual, observar prazo e instruções de entrega no moodle

Arquivos necessários

1. 42049.png [disponível no zip; fonte:
<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/BSDS300/html/dataset/images/gray/42049.html>]

Converting Between Image Classes

<http://www.mathworks.com/help/images/convert-image-data-between-classes.html>

Image Type Conversion

<http://www.mathworks.com/help/images/image-type-conversions.html>

a) Classes de dados e conversões para a representação de imagens

Rode o código e observe os resultados. O objetivo é aprender quais são as classes de dados e as conversões mais usadas no Octave para a representação de imagens. Não é necessário entregar.

```
%Convertendo uma imagem de entrada uint8
clear all, close all, clc
a = [-3 0; 0.25 63.6; 0.5 270.9]

%Simplesmente trunca entre [0 255] e os não inteiros
%são arredondados para o inteiro mais próximo
b = uint8(a)

%im2double
%http://www.mathworks.com/help/images/ref/im2double.html
%I2 = im2double(I) converts the intensity image I to double precision,
%rescaling [importante saber o que é esse 'rescaling'] the data
%if necessary. If the input image is of class double, the output
%image is identical.
%Para o Octave, uma imagem 'double precision' está na faixa [0 1].
%Como a imagem de entrada é uint8, a im2double faz o 'rescaling' para
%adaptar para a faixa [0 1]: 0->0, 255->1, apenas dividindo todos
%os valores por 255.
c = im2double(b)

%Adapta para a faixa [0 255]: 0->0, 255->255
d = im2uint8(b)

%Convert matrix to grayscale image
%http://www.mathworks.com/help/toolbox/images/ref/mat2gray.html
%minimo(b)->0, maximo(b)->1. Isto é chamado de 'normalização' ou 'ajuste
%de contraste'
g = mat2gray(b)

bd = double(b) %para fazer operações sobre 'b' em double precision
g2 = ((bd-min(bd(:))) / (max(bd(:))-min(bd(:))))*1 %igual ao mat2gray(b)
```

b) Classes de dados e conversões para a representação de imagens

Rode o código e observe os resultados. O objetivo é aprender quais são as classes de dados e as conversões mais usadas no Octave para a representação de imagens. Não é necessário entregar.

```
%Conversões sobre uma imagem de entrada double
clear all, close all, clc
a = [-3 0; 0.25 63.6; 0.5 270.9]

%im2double
%http://www.mathworks.com/help/toolbox/images/ref/im2double.html
%I2 = im2double(I) converts the intensity image I to double precision,
%rescaling the data if necessary. If the input image is of class double,
%the output image is identical.
%Como a entrada é double, a saída é idêntica.
c = im2double(a)

%Como a entrada é double, considera que a matriz de entrada tem valores
%na faixa [0 1] e faz 0->0, ... , 0.25->64, ... , 0.5->127, ... ,
%0.75->191, ... , 1->255.
%Valores menores ou iguais 0 -> 0, maiores ou iguais 1 -> 255
d = im2uint8(a)

%Convert matrix to grayscale image
%http://www.mathworks.com/help/toolbox/images/ref/mat2gray.html
%minimo(a)->0, maximo(a)->1. Isto é chamado de 'normalização' ou 'ajuste
%de contraste'
g = mat2gray(a)
g2 = (a-min(a(:))) / (max(a(:))-min(a(:))) * 1 %igual ao mat2gray(a);
```

1.1) Logical indexing

Para a imagem *42049.jpg*: substituir todos os pixels de valor menor que 20 por 20 e os maiores que 200 por 200. Salvar a nova imagem como *42049_20-200.png*, para utilização nos próximos exercícios.

Veja o script *a02_logicalIndexing.m*. Ele apresenta um template para te ajudar neste exercício.

Nome do .m: *atv02_01*

1.2) Classes de dados e conversões para a representação de imagens, visualização (imshow)

```
%a02_02 [script]
clear all; close all; clc

g=imread('42049_20-200.png');
mg=min(g(:)); Mg=max(g(:));
figure, imshow(g), title('g')
g1=double(g); mg1=min(g1(:)); Mg1=max(g1(:));
figure, imshow(g1), title('g1')
g2=im2double(g); mg2=min(g2(:)); Mg2=max(g2(:));
figure, imshow(g2), title('g2')
g3=mat2gray(g); mg3=min(g3(:)); Mg3=max(g3(:));
figure, imshow(g3), title('g3')
```

```
% Responda aqui no próprio script
% g é da classe:
% Mín em g: , Máx em g:
% g1 é da classe:
% Mín. em g1: , Máx em g1:
% imshow(g1) mostra uma imagem:
% porque:
% g2 é da classe:
% Mín em g2: , Máx em g2:
% g3 é da classe:
% Mín em g3: , Máx em g3:
% Qual é a diferença entre as funções
% im2double e mat2gray?
```

Rode o código e responda as questões no próprio script.

Nome do .m: atv02_02

1.3) Visualização (imread, imshow, subplot)

Usar a função *subplot* para mostrar a imagem *42049_20-200.png* original e com autocontrast (função *mat2gray*) em uma mesma figure, uma ao lado da outra. Para inserir os nomes de cada imagem na figure, utilize a função *title* logo após a função *imshow*. O nome da primeira deve ser 'Original' e da segunda 'Autocontrast'.

Nome do .m: atv02_03

1.4) Indexação da matriz imagem (imtool, operador colon, imwrite)

Criar, mostrar e salvar uma imagem no formato *png*, contendo apenas o pássaro da imagem *42049_20-200.png*. Também mostrar a *42049_20-200.png* com um quadrado preto sobre a área que foi recortada. Dica: a primeira coisa a fazer é obter as coordenadas do canto superior direito e do canto inferior esquerdo do pássaro na imagem *42049_20-200.png*.

Veja o script *a02_submatriz.m*. Ele apresenta um template para te ajudar neste exercício.

Nome do .m: atv02_04

1.5) Operador colon para criar vetores e função repmat

Criar, mostrar e salvar uma imagem *png* de dimensão *height = 50 pixels*, contendo toda a escala de níveis de cinza de resolução 8 bits por pixel, em ordem crescente, conforme mostrado a seguir (se desejar, crie tb as bordas pretas de 1 pixel). Dica: usar o operador *colon* e a função *repmat*.



Veja o script *a02_graysUint8*. Ele apresenta um template para te ajudar neste exercício.

Nome do .m: atv02_05

1.6) Função unique

42049.jpg é de 8 bits por pixel (uint8). Portanto, pode ter até 256 níveis de cinza diferentes... mas não tem. Quantos níveis de cinza diferentes há em *42049.jpg*? Dica: usar a função *unique*.

Nome do .m: atv02_06