Instituto **S**uperior de **E**ngenharia de **L**isboa

ENGENHARIA INFORMÁTICA E DE COMPUTADORES

PROCESSAMENTO DE IMAGEM E BIOMETRIA

PRIMEIRO TRABALHO PRÁTICO

LI61N-MI2N

Autores – Grupo 3:

40602, Sara Sobral

40686, Eduardo António

Índice

Desenvolvimento de conclusões

Exercício 1

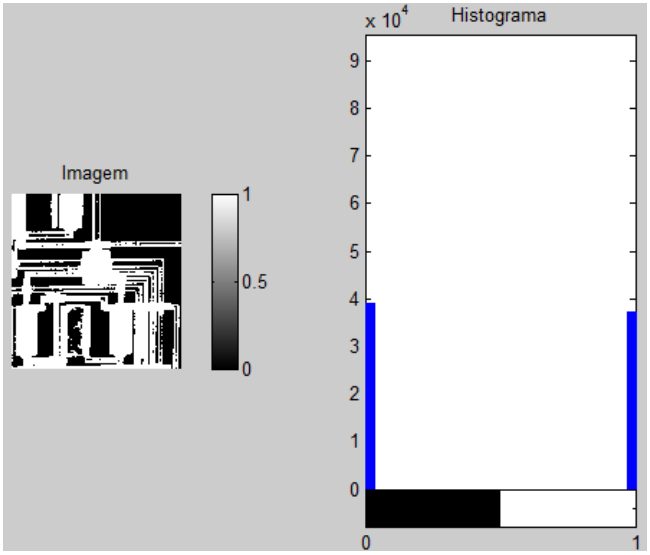
Função image_details.m

- **Obtenha informação sobre uma imagem. espacial; resolução em profundidade; valores mínimo, médio e máximo de intensidade; medida de contraste; entropia da imagem. Apresentar a imagem e o respetivo histograma.**

Utilização da função imfinfo para obter informação sobre a imagem, como a resolução espacial, a resolução em profundidade e o tipo de cor da imagem para saber se é necessário obter as componentes (R, G e B).
Utilização da função imread para ler a imagem a partir do ficheiro.
A função recebe como parâmetro a imagem que se pretende analisar. As imagens estão presentes em GenericImages.zip.

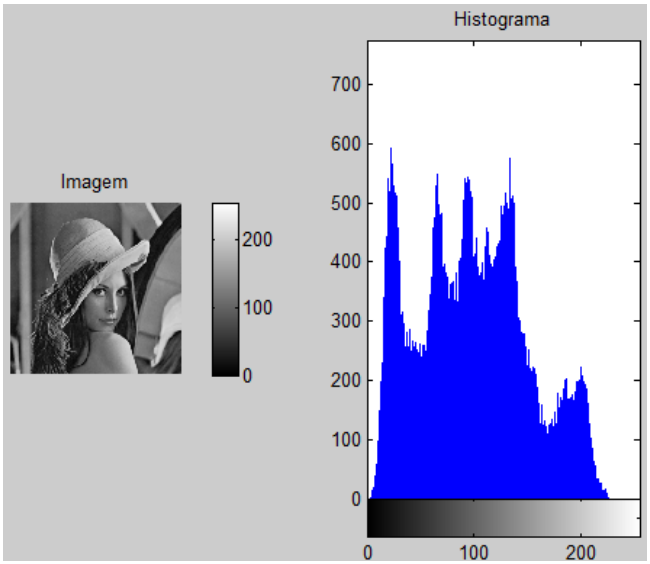
- **Apresentação de resultados**

- Exemplo de imagem binária: circ_bw.tif



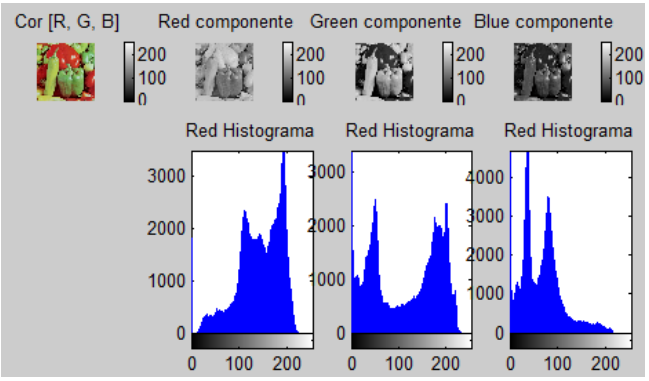
resolução espacial = 76160
resolução em profundidade = 1 bit/pixel
valores mínimo de intensidade = 0
valores médio de intensidade = 0.48871
valores máximo de intensidade = 1
medida de contraste = 6.0206
entropia da imagem = 0.99963

- Exemplo de imagem monocromática: lena.gif



resolução espacial = 65535
resolução em profundidade =
8 bit/pixel
valores mínimo de intensidade = 3
valores médio de intensidade = 99
valores máximo de intensidade = 238
medida de contraste = 35.5268
entropia da imagem = 7.5683

- Exemplo de imagem a cores: peppers.png



resolução espacial = 262144
resolução em profundidade = 24
bit/pixel

componente	red:	green:	blue:
valores mínimo de intensidade =	0	0	0
valores médio de intensidade =	144.1151	112.8049	64.1288
valores máximo de intensidade =	253	255	255
medida de contraste =	48.0967	48.1308	48.1308
entropia da imagem =	7.3316	7.5605	7.0196

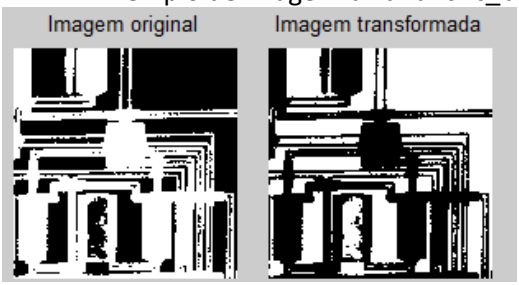
Função image_details.m

- **Obtenha informação sobre uma imagem e a uma versão transformada por T, sendo T uma transformação de intensidade genérica.**

A função recebe como parâmetro a imagem que se pretende analisar. As imagens estão presentes em GenericImages.zip.
Função T utilizada foi a inversa, ou seja, o índice 0 da *lookup table* corresponde à intensidade mais elevada que a imagem pode ter.
Utilização da função intlut para gerar a imagem transformada.

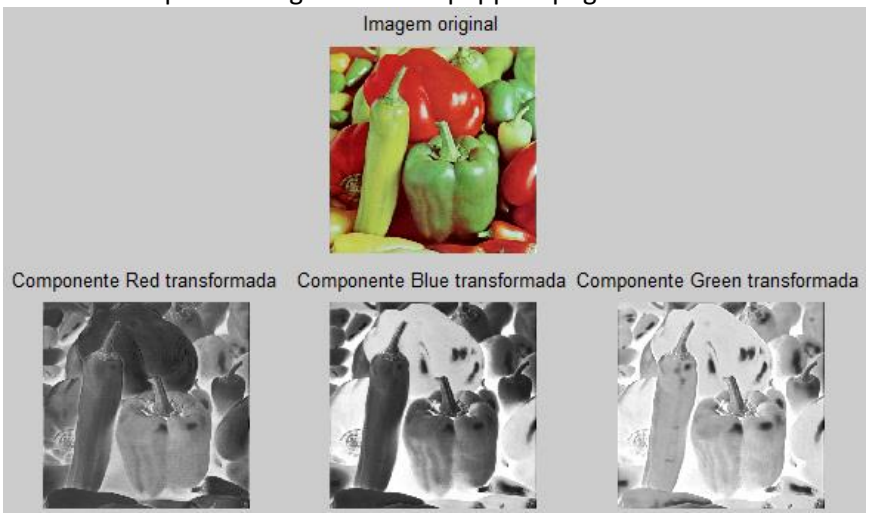
- **Apresentação de resultados**

- Exemplo de imagem binária: circ_bw.tif



- Exemplo de imagem monocromática: lena.gif

- Exemplo de imagem a cores: peppers.png



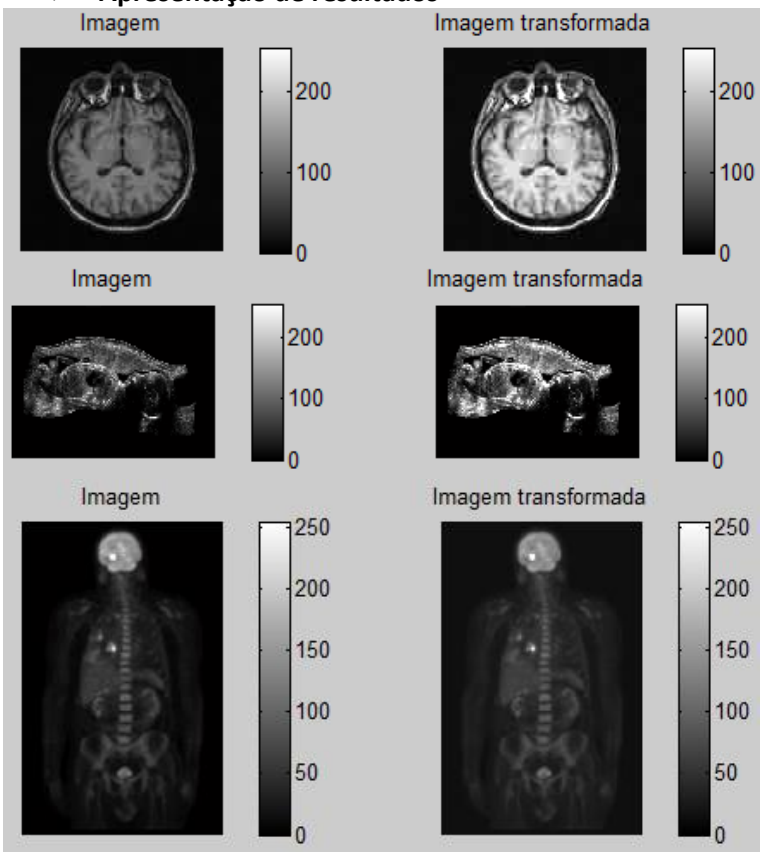
Exercício 2

Função medical_image_enhancement.m

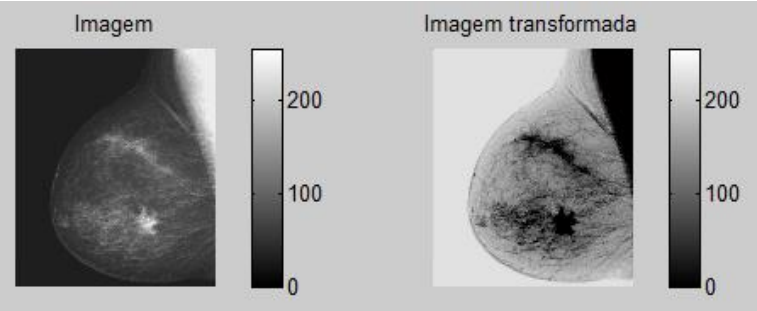
- **Realiza transformações de intensidade adequadas para melhorar a legibilidade das mesmas. Apresente os resultados obtidos para cada imagem deste conjunto.**

A função recebe como parâmetro a imagem que se pretende analisar. As imagens estão presentes em MedicalImages.zip. Se a imagem recebida for binária ou coloridas é convertida para níveis de cinzento para que se possa aplicar a função imadjust. Esta função ajusta os valores de intensidade da imagem, ou seja, mapeia os valores de intensidade da imagem para novos valores de forma a que 1% dos dados sejam saturados em baixas e altas intensidades. O que faz aumentar o contraste da imagem de saída..

- **Apresentação de resultados**



No caso de PET1.tif obtemos os tumores depois o contorno do corpo, a apresentamos a sua soma.



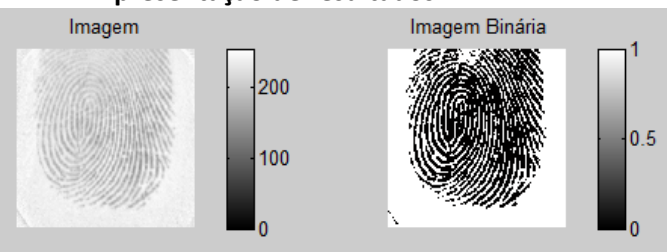
No caso XRay1.tif obtemos a inversa à qual subtraímos a imagem original.

Função fingerprint_enhancement.m

- **Para uma imagem de impressão digital produz uma versão binária da mesma, tentando separar as riscas do fundo.**

A função recebe como parâmetro a imagem que se pretende analisar. As imagens estão presentes em FingerprintImages.zip. Nesta função calcula-se o limiar ótimo para transformar a imagem na sua versão binária (método de Otsu) através da função im2bw..

- **Apresentação de resultados**



Função face_detection.m

- **Para uma imagem de face, procura localizar os extremos da face e afixar um retângulo a delimitar a face.**

A função recebe como parâmetro a imagem que se pretende analisar. As imagens estão presentes em FaceImages.zip. Foi utilizado um detetor de objetos (https://www.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-class.html) que utiliza o algoritmo Viola-Jones. É criado um detetor de objetos com classificação modelo 'FrontalFaceCART', utliza-se a função step que processa os dados de entrada de acordo com o algoritmo do objeto e retorna uma matriz [x y largura altura], que especifica em pixels, o canto superior esquerdo e o tamanho de uma caixa delimitadora.

- **Apresentação de resultados**

Exercício 3

- **Identifique o(s) problema(s) na imagem e proponha uma técnica (ou mais) para a sua correção;**

- circles.bmp

Problema(s): As imagens apresentam ruido padronizado com diferentes níveis de intensidade.

Imagens	**Proposta(s) de correção**
circles_1.bmp	Aplicar uma transformação em frequência passa baixo para remover o ruido que não pertence ao espetro da imagem e aplicar um filtro de *sharpening* para salientar as transições entre os círculos da aplicação anterior.
circles_2.bmp	
circles_3.bmp	
circles_4.bmp	
circles_5.bmp	

- face1.bmp

Problema(s): As imagens apresentam diferentes intensidades de ruido impulsivo salt and peppe.

Imagens	**Proposta(s) de correção**
face1_1.bmp	Aplicar um filtro espacial não linear mediana para eliminar pontos de ruido impulsivo com valores extremos. A dimensão da mascara mediana pode variar para corrigir as diferentes imagens. Obriga à ordenação dos valores, colocando os valores 0 (pepper) à esquerda, os valores 255 (salt) à direita e os valores dos pixéis originais no centro, obtendo assim um valor do contexto da imagem original.
face1_2.bmp	
face1_3.bmp	
face1_4.bmp	
face1_5.bmp	

- finger1.bmp

Problema(s): As imagens estão esborratadas, o nível de *blured* vai aumentando de imagem para imagem.

Imagens	**Proposta(s) de correção**
finger1_1.bmp	Aplica rum filtro espacial linear passa alto (como o laplaciano) para realizar *sharpening* sobre a imagem, ou seja, afiar e salientar as transições das zonas brancas para as pretas.
finger1_2.bmp	
finger1_3.bmp	
finger1_4.bmp	
finger1_5.bmp	

- lena.gif

Problema(s): Todas as imagens apresentam ruido.

Imagens	**Proposta(s) de correção**
lena_1.bmp	Aplicar um filtro espacial linear passa baixo para suavizar a imagem de forma a eliminar o ruido existente.
lena_2. bmp	
lena_3. bmp	
lena_4.bBmp	
lena_5. bmp	

- squares.gif

Problema(s): As imagens estão esborratadas, o nível de *blured* vai aumentando de imagem para imagem.

Imagens	**Proposta(s) de correção**
squares_1.gif	Aplicar um filtro espacial passa alto de deteção de contornos.
squares_2.gif	
squares_3.gif	
squares_4.gif	

- **Compare a imagem restaurada com a imagem original.**

Um MAE baixo e um MSE alto indica ocorrência de outliers no conjunto de dados.

Imagens	Brilho	Contraste	Entropia	MSE	MAE
circles_1.bmp	4.4514	0	-2.8446	0.65889	4.444
circles_2.bmp	2.9665	0	-2.5703	0.70064	2.9599
circles_3.bmp	0.61072	0	-1.4036	0.98042	0.60997
circles_4.bmp	0.61281	0	-1.4109	0.97893	0.61208
circles_5.bmp	0.62746	0	-1.4329	0.97481	0.62737
face1_1.bmp	0.077698	0	0.066425	0.46234	1.6896
face1_2.bmp	0.078674	0	0.060538	0.45258	1.7401
face1_3.bmp	0.11615	0	0.06493	0.44232	1.8273
face1_4.bmp	0.10431	0	0.06076	0.41498	1.8867
face1_5.bmp	0.11865	0	0.062689	0.41351	2.1005
finger1_1.bmp	-3.3247	0	0.14186	0.4341	5.8699
finger1_2.bmp	-2.5914	0	0.17585	0.14763	8.3205
finger1_3.bmp	-1.8762	0	0.22112	0.069748	11.5573
finger1_4.bmp	-1.1661	0	0.252	0.04567	14.8908
finger1_5.bmp	-0.46469	0	0.25703	0.032974	0.032974
lena_1.gif	0.31051	0	0.029094	1	14
lena_2.gif	0.31051	0	0.029094	1	14
lena_3.gif	0.31051	0	0.029094	1	14
lena_4.gif	0.31051	0	0.029094	1	14
lena_5.gif	0.31051	0	0.029094	1	14
squares_1.gif	9.7495	0	-4.5947	0	11
squares_2.gif	-10.1055	0	-4.0995	0	11
squares_3.gif	-10.2864	0	-3.7359	0	11
squares_4.gif	-10.4003	0	-3.4807	0	11
squares_5.gif	-10.483	0	-3.2985	0	11

Exercício 4

Função codeCardGenerato

- **A qual gera uma imagem colorida, contendo um cartão de códigos, com conteúdo aleatório, de forma matricial, tal como se apresenta na figura. Apresente cinco imagens diferentes geradas com o método proposto.**

- **Descrição da metodologia**

Este exercício tem como objetivo gerar uma imagem colorida com conteúdo aleatório na forma matricial.

1. Gerar um valor (inteiro) aleatório para a largura N e um valor (caracter) para altura M.
2. Adicionar a um contento os valores de 1 a N com espaçamento de 3.
3. Criar um objeto imagem (*Graphics2D*). Ter cuidado com o tipo de fonte para que o espaçamento não fique desorganizado.
4. Gerar um valor aleatório para a escolha da cor, a cor é introduzida por linha.
5. Escrever no objeto imagem o conteúdo do contentor com a cor gerada.
6. Gerar M linhas com N valores (inteiros) aleatórios.

7. Cada linha gerada é adicionada ao contentor, é gerada também uma cor aleatória. Repetir o ponto 5.
8. Preencher o fundo da imagem com uma cor.

- **Apresentação de resultados**

	1	2	3	4	5	6	7	8
A	506	532	344	227	481	517	514	17
B	420	220	714	309	974	765	329	66
C	538	882	311	622	658	781	846	208
D	251	294	311	679	848	378	283	157
E	556	462	455	358	947	564	793	148
F	609	403	947	77	392	187	185	494
G	838	175	990	954	226	392	321	196
H	61	435	729	172	571	930	47	931
I	838	983	104	170	426	206	35	950

Exercício 5

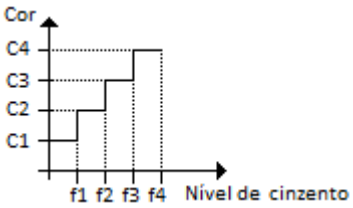
As imagens utilizadas estão presentes em BinaryAndGrayscaleImages.zip.

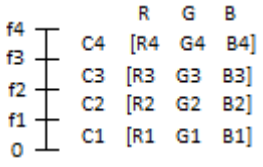
- **Realize coloração das imagens através das técnicas de intensity slicing e intensity to RGB transform; indique os critérios e as funções usadas para a atribuição de cores.**

As técnicas recebem imagens monocromáticas e realização a sua coloração. É útil para visualizar imagens médicas / científicas / vegetação, pois é de interesse realçar certos valores de intensidade para serem mais percetíveis ao sistema visual humano.

A técnica de intensity slicing começa por dividir a resolução em profundidade da imagem recebida por um valor (escala) obtendo o número de intervalos. O valor escala é calculado consoante os valores de níveis de cinzento que a imagem original usa. Logo, o número de intervalos é igual ao número de cores da imagem original que por sua vez é igual ao número de cor que a imagem final vai ter.
Para cada intervalo é construído um valor RGB. No final a imagem monocromática de dimensão $(2^n - 1)^2$passa para $(2^n - 1)^3$.
Exemplo:
Imagem monocromática (circles.bmp) com resolução em profundidade n = 8 bit/pixel

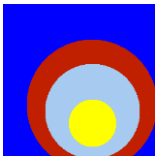
A escala é 4 porque a imagem apenas usa quatro níveis cinzentos.



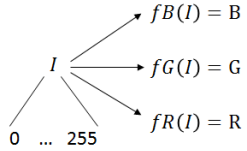


$$f(x) = \begin{cases} C1, 0 \leq x \leq 63 \\ C2, 64 \leq x \leq 127 \\ C3, 128 \leq x \leq 191 \\ C4, 192 \leq x \leq 255 \end{cases}$$

Quando o valor do pixel está mais perto de 0 a cor RGB atribuída tem um tom frio, e quando o valor do pixel está mais perto de 255 a cor RGB atribuída tem um tom quente. Foi utilizada uma tabela de lookup.
Resultado:



A técnica intensity to RGB transform aplica três funções diferentes sobre a imagem monocromática recebida. Cada função gera a componente azul, a componente verde e a componente vermelha.



- **Comente qual das técnicas aplicadas produz melhores resultados.**

A técnica de intensity slicing apresenta melhor resultados, pois se a função escolhida na técnica intensity to RGB transform não for adequada pode fazer o oposto de salientar detalhes.
Exemplo:
intensity slicing: intensity to RGB transform