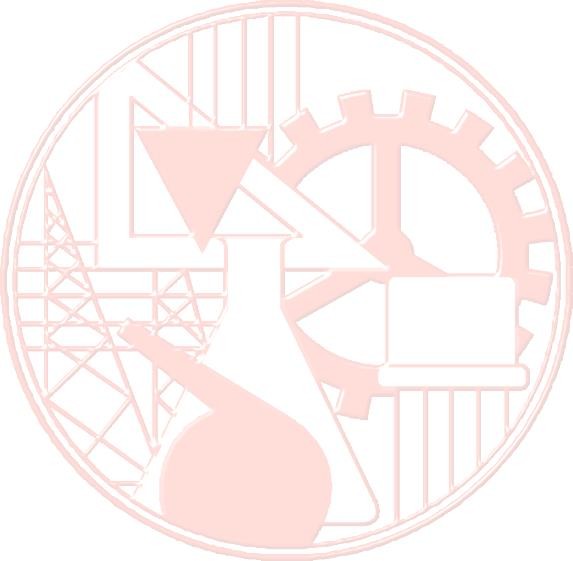
**I**nstituto **S**uperior de **E**ngenharia de

**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 3](#_bookmark0)

[Exercício 1 3](#_bookmark1)

[Recorrendo à função file\_entropy.m 3](#_bookmark2)

[Recorrendo à função coder\_evaluation.m 6](#_bookmark3)

[Exercício 2 7](#_bookmark4)

[Recorrendo à função coder\_decoder\_evaluation.m 7](#_bookmark5)

[Exercício 3 8](#_bookmark6)

[Exercício 4 9](#_bookmark7)

[REGRAS DO SISTEMA DE INFORMAÇÃO 10](#_bookmark8)

[Exercício 5 10](#_bookmark9)

# Desenvolvimento de conclusões

## Exercício 1

### Função image\_details.m

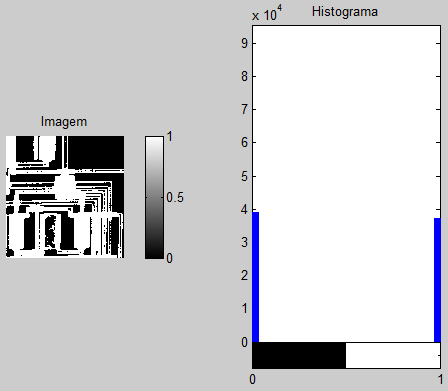
* **Obtenha informação sobre uma imagem. espacial; resolução em profundidade; valores mínimo, médio e máximo de intensidade; medida de contraste; entropia da imagem. Apresentar a imagem e o respetivo histograma.**

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

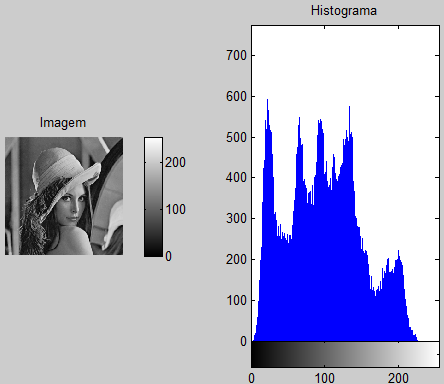
valores médio de intensidade = 0.48871

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

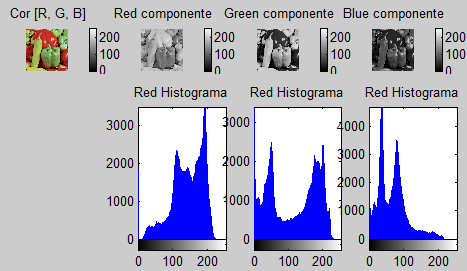
valores mínimo de intensidade = 3

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

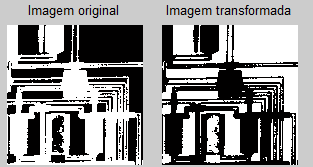
* **Obtenha informação sobre uma imagem e a uma versão transformada por T, sendo T uma transformação de intensidade genérica.**

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.

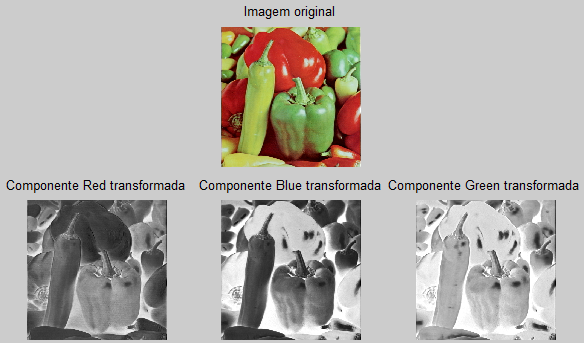
* **Apresentação de resultados**
* Exemplo de imagem binária: circ\_bw.tif



* Exemplo de imagem monocromática: lena.gif



* Exemplo de imagem a cores: peppers.png



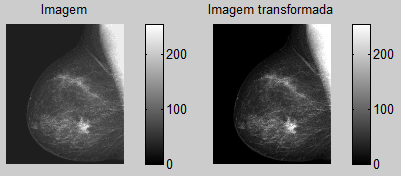
## Exercício 2

### 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**

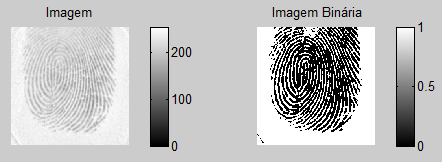


### Função fingerprint\_enhancement.m

* **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

|  |  |  |
| --- | --- | --- |
| Imagens | Problemas | Proposta(s) de correção |
| circles\_1.bmp | As imagens apresentam ruido padronizado com diferentes níveis. | Transformação de frequência passa baixo |
| circles\_2.bmp |
|  |
|  |
|  |
| circles\_3.bmp |  | Transformação de frequência passa baixo + sharpening |
| circles\_4.bmp |  |
| circles\_5.bmp |  |

face1.bmp

|  |  |  |
| --- | --- | --- |
| Imagens | Problemas | Proposta(s) de correção |
| face1\_1.bmp | As imagens apresentam ruido impulsivo salt and Pepper com diferente intensidade. | Mediana |
| face1\_2.bmp |
| face1\_3.bmp |
| face1\_4.bmp |
| face1\_5.bmp |

finger1.bmp

|  |  |  |
| --- | --- | --- |
| Imagens | Problemas | Proposta(s) de correção |
| finger1\_1.bmp | As imagens estão esborratadas, umas apresentação um esborratamento mais acentuado que outras. | Transformação de frequência passa alto + binarizaçao |
| finger1\_2.bmp |
| finger1\_3.bmp |
| finger1\_4.bmp | Sharpening passa alto + binarização |
| finger1\_5.bmp |

lena.gif

|  |  |  |
| --- | --- | --- |
| Imagens | Problemas | Proposta(s) de correção |
| lena\_1.gif |  |  |
| lena\_2.gif |
| lena\_3.gif |
| lena\_4.gif |
| lena\_5.gif |

squares.gif

|  |  |  |
| --- | --- | --- |
| Imagens | Problemas | Proposta(s) de correção |
| squares\_1.gif | Imagem esborratada | LoG + binarização |
| squares\_2.gif | Sharpening + binarização |
| squares\_3.gif |
| squares\_4.gif |

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

Comparar gif com bmp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Imagens | Brilho | Contraste | Entropia | MSE | MAE |
| circles\_1.bmp |  |  |  |  |  |
| circles\_2.bmp |  |  |  |  |  |
| circles\_3.bmp |  |  |  |  |  |
| circles\_4.bmp |  |  |  |  |  |
| circles\_5.bmp |  |  |  |  |  |
|  |  |  |  |  |  |
| face1\_1.bmp | 1.1616 | 0 | 0.066425 | 0.46234 | 0 |
| face1\_2.bmp | 1.1394 | 0 | 0.060538 | 0.45258 | 0 |
| face1\_3.bmp |  |  |  |  |  |
| face1\_4.bmp |  |  |  |  |  |
| face1\_5.bmp |  |  |  |  |  |
|  |  |  |  |  |  |
| finger1\_1.bmp |  |  |  |  |  |
| finger1\_2.bmp |  |  |  |  |  |
| finger1\_3.bmp |  |  |  |  |  |
| finger1\_4.bmp |  |  |  |  |  |
| finger1\_5.bmp |  |  |  |  |  |
|  |  |  |  |  |  |
| lena\_1.gif |  |  |  |  |  |
| lena\_2.gif |  |  |  |  |  |
| lena\_3.gif |  |  |  |  |  |
| lena\_4.gif |  |  |  |  |  |
| lena\_5.gif |  |  |  |  |  |
|  |  |  |  |  |  |
| squares\_1.gif |  |  |  |  |  |
| squares\_2.gif |  |  |  |  |  |
| squares\_3.gif |  |  |  |  |  |
| squares\_4.gif |  |  |  |  |  |
| squares\_5.gif |  |  |  |  |  |

## 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**



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