

**Instituto Superior de Engenharia de
Lisboa**

**ENGENHARIA INFORMÁTICA E DE
COMPUTADORES**

PROCESSAMENTO DE IMAGEM E BIOMETRIA

SEGUNDO TRABALHO PRÁTICO

LI61N-MI2N

Autores – Grupo 3:

40602, Sara Sobral

40686, Eduardo António

Índice

Desenvolvimento de conclusões	3
Exercício 1.....	3
Função coloring_medical_images.m.....	3
Exercício 2.....	4
Função rgb2safeColor.m	4
Exercício 3.....	5
Função fingerprint_enhancement_morph.m	5
Função minutiae_detection.m	5
Exercício 4.....	6
Projete, realize e avalie um módulo de deteção e reconhecimento facial, para um número reduzido de utilizadores	6

Desenvolvimento de conclusões

Exercício 1

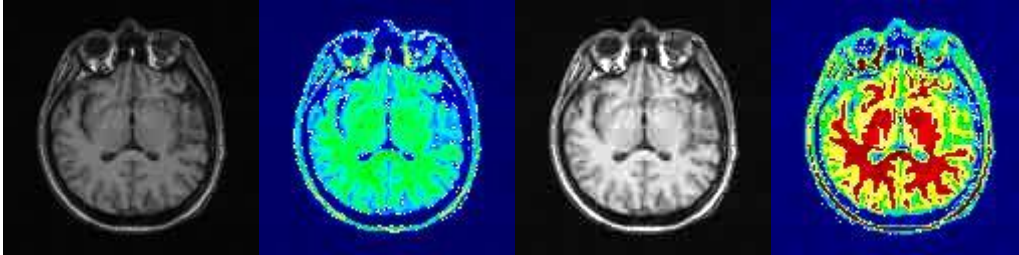
Função `coloring_medical_images.m`

- Realize a coloração das imagens médicas originais e das imagens produzidas pela função `medical_image_enhancement.m`, recorrendo a uma técnica de coloração à escolha.

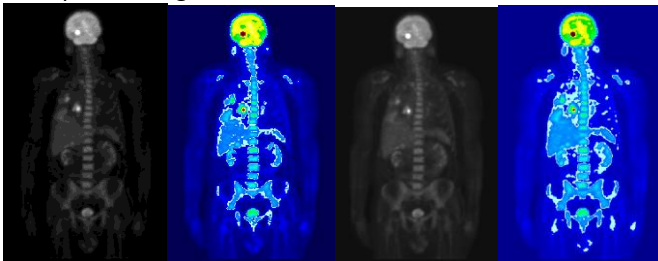
A técnica escolhida foi a intensity slicing.

- Apresentação de resultados

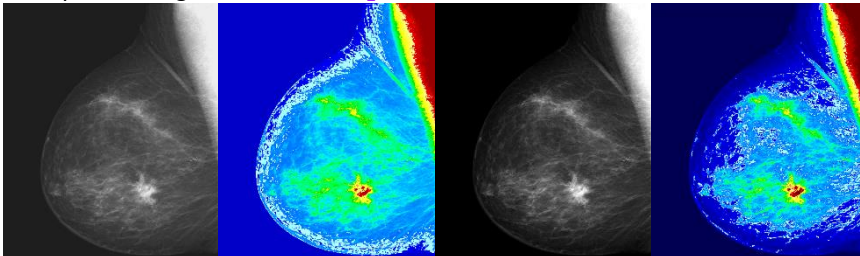
Exemplo de imagem binária: [MR1.jpg](#)



Exemplo de imagem binária: [PET1.tif](#)



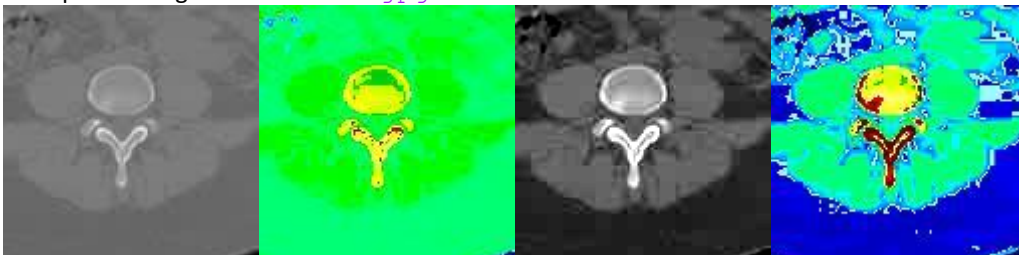
Exemplo de imagem binária: [XRay1.tif](#)



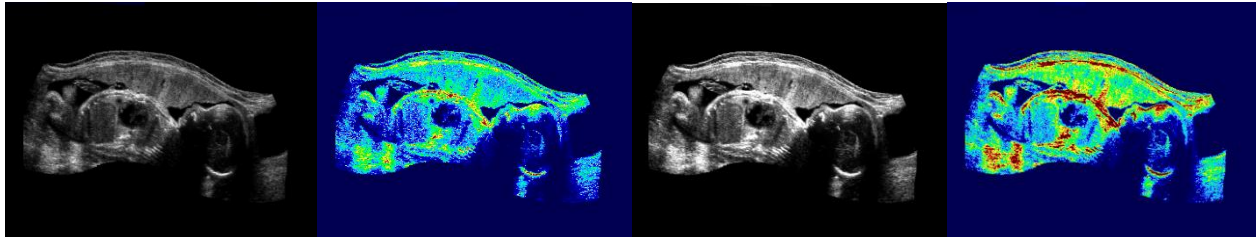
Exemplo de imagem binária: [XRay2.tif](#)



Exemplo de imagem binária: [CT1.jpg](#)



Exemplo de imagem binária: [US1.tif](#)



Em todos os exemplos é notável a diferença entre os resultados da coloração da imagem original e da coloração da imagem produzida pela função `medical_image_enhancement.m`. Esta última quando colorida apresenta mais detalhes visíveis ao ao plho humano.

Exercício 2

Função `rgb2safeColor.m`

- Converta uma imagem RGB na sua versão web safe color ou web safest color, mediante um parâmetro de entrada, retornando o número de cores distintas da imagem resultante.

Para a comparação das competes com as safe colors foi gerado um vetor com os valores possíveis de uma componente [0, 51, 102, 153, 204, 255]. O valor de cada pixel de cada componente é comparado através da distancia Eucladiana com os possíveis valores do vetores, o que apresentar distancia menor é o novo valor do pixel da componente.

	0	3	6	9	C	F
00	*000*					*F00*
03	*003*					*F03*
0F	*00F*					*F0F*
F0	*0F0*		*6F0*			*FF0*
F3		*3F3*	*6F3*			*FF3*
F6	*0F6*	*3F6*			*CF6*	*FF6*[12]
FC	*0FC*	*3FC*				
FF	*0FF*	*3FF*	*6FF*			*FFF*

Key	Hex	Decimal	Fraction
0	00	0	0
3	33	51	0.2
6	66	102	0.4
9	99	153	0.6
C or (12)	CC	204	0.8
F or (15)	FF	255	1

Para a comparação das competes com as safest colors foi gerado um array {22x3} com as combinações possíveis de cores. O valor de cada pixel de cada componente (no formato vetor [R,G,B]) é comparado através da distancia Eucladiana com cada linha do array, o que apresentar distancia menor é o novo valor do pixel de cada componente.

- Apresentação de resultados

Exemplo: [codeCard2.png](#)

Imagem original	Web safe colors	Web safest colors
Distinct colors:	14	7

Exemplo: [codeCard1.png](#)

Imagem original	Web safe colors	Web safest colors
Distinct colors:	11	6

Os resultados obtidos dependem das cores da imagem de entrada, pois consoante a cor é calculada a cor *safe* ou *safest* mais próxima. Visualmente as imagens mudam mais com as *web safest color* do que com as *web safe color* devido à quantidade de cores disponíveis para cada versão, logo existe mais perda na versão *web safest color*.

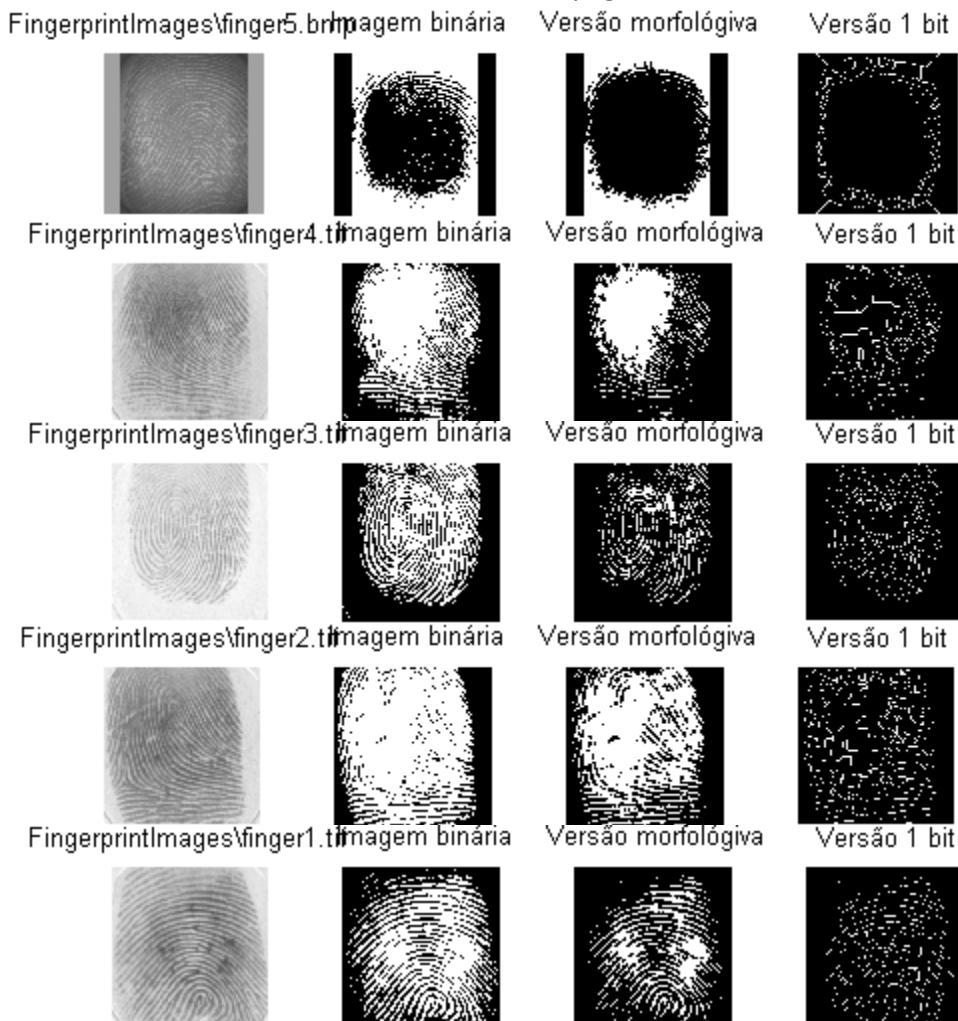
Exercício 3

Função `fingerprint_enhancement_morph.m`

A primeira imagem é a imagem original. A segunda imagem é a sua versão binária com um limiar ótimo aplicando o método de Otsu. A terceira imagem é a versão morfológica, obtida aplicando sobre a imagem binária as operações de abertura seguida de erosão. A quarta imagem é o esqueleto da versão morfológica, os pixels nos limites são removidos, mas as linhas mantêm ligadas, as linhas da impressão digital apenas têm 1 bit de grossura.

➤ Apresentação de resultados

NOTA: A qualidade das imagens não é a melhor por isso as imagens estão disponíveis na pasta do exercício 3 com os nomes 1, 2, 3, 4 e 5 no formato .png.



Função `minutiae_detection.m`

Esta função percorrer o esqueleto da versão morfológica, a qual resulta da função anterior, com as seguintes máscaras de forma a encontrar bifurcamentos.

<div> <div>0</div><div>1</div><div>0</div> <div>0</div><div>1</div><div>0</div> <div>1</div><div>0</div><div>1</div> </div>	Se o pixel central é 1 e tem exatamente 3 vizinhos de valor um, então o pixel central é um ramo.
<div> <div>0</div><div>0</div><div>0</div> <div>0</div><div>1</div><div>0</div> <div>0</div><div>0</div><div>1</div> </div>	Se o pixel central for 1 e tiver apenas 1 vizinho de valor um, então o pixel central é uma extremidade.

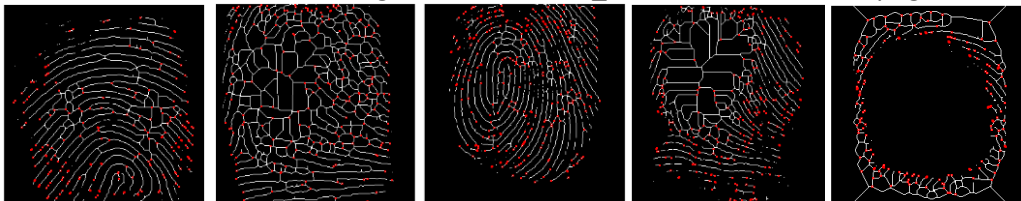
0	1	0
0	1	1
1	0	0

Se o pixel acima e o pixel à direita têm valor 1, então o ramo é triplo.

Todos os pixels onde esses bifurcamentos acontecem são guardados numa nova imagem binária que servirá de máscara. Fazer uma soma da imagem (no formato RGB) com a máscara criada para obter os bifurcamentos com uma cor distinta.

➤ Apresentação de resultados

NOTA: A qualidade das imagens não é a melhor por isso as imagens estão disponíveis na pasta do exercício 3 com os nomes 1, 2, 3, 4 e 5 seguido de minutiae_detection no formato .png.



Apesar de alguns bifurcamentos não serem detetados ao realizar esta operação é muito mais perceptível todos as terminações de ramos e todos os bifurcamentos. Através destes ponto torna-se mais fácil comparar impressões digitais.

Exercício 4

Projete, realize e avalie um módulo de detecção e reconhecimento facial, para um número reduzido de utilizadores

➤ Descrição da realização das funcionalidades de detecção e de reconhecimento.

Foi criada uma aplicação em c# com as seguintes opções:

1. Extrair características das:
 - a. Imagens fornecidas na pasta FacelImages
 - b. Imagens dos elementos do grupo
 - c. Imagens fornece Cidas na base de dados CaltechFaceDatabase
2. Correr um classificador identificando a diretoria e o nome do ficheiro da imagem

Foi criada um aplicação em matlab (a partir da demo14) com a seguinte funcionalidade:

1. Utilizar o software weka para treinar e testar o classificador e avaliar a sua performance
 - a. Obtendo uma tabela de comparação de resultados, a percentagem de erro global e o número de vezes em que o classificador falhou

Aplicação c#

Para **extrair as características** de uma pessoa é utilizada a Microsoft Face API, disponibilizada pela Microsoft AZURE em <https://docs.microsoft.com/pt-pt/azure/cognitive-services/face/quickstarts/csharp>. Esta API apenas permite 20 pedidos por minuto por isso de 20 em 20 pedidos é feito uma espera de 1 minuto e meio. A chave deste tipo de aplicações é encontrar áreas de contrastes. O algoritmo de detecção é conhecido por Viola-Jones, funciona por repetidamente analisar as diferenças dos valores dos pixels numa escala. Por exemplo a cana do nariz é mais clara do que a zona que a rodeia. Se forem encontradas suficientes compatibilidades chega-se à conclusão que está ali uma cara. Para detetar características faciais é usado um modelo baseado em amostras de coordenadas, as coordenadas são ajustadas às características da pessoa na imagem. Com as coordenadas da imagem é criada um objeto em json.

As características (que além de pontos de referia da máscara podem também ser detalhes como a cor do cabelo ou o tipo de óculos) estas são ordenadas em vetores. **Neste caso apenas foi utilizada uma característica (a idade) pois é suficientemente distinta, discriminatória e não tem dependências.**

O classificador foi gerado pelo software weka, na aplicação matlab. O reconhecimento é feito extraindo as características da pessoa na imagem e é apresentado o resultado do classificador.

Aplicação matlab:

Com os vetores obtidos é organizada uma matriz com o seguinte formato:

$$\begin{bmatrix} característica_1, \dots, característica_n, etiqueta_1 \\ \vdots \\ característica_1, \dots, característica_n, etiqueta_m \end{bmatrix}$$

Esta matriz é processada pela aplicação usando o software weka. Se a margem de erro global for aceitável regista-se classificar em árvore gerado na aplicação C#, caso contrário repete-se tudo a partir do ponto 2.

- **Apresentação de resultados experimentais de deteção e reconhecimento, usando imagens de uma base de dados específica para este efeito ou do conjunto FacelImages.zip.**

Matriz construída sobre os vetores obtidos pela aplicação C#

```
[42.9,1; 45.2,1; 44.8,1; 45.2,1; 44.5,1; 43.3,1; 40.1,1; 43.0,1; 45.1,1;
46.6,1; 43.6,1; 44.6,1; 43.3,1; 45.8,1; 44.8,1; 46.7,1; 45.4,1; 43.4,1;
47.8,1; 44.5,1; 45.6,1;
36.8,2; 36.1,2; 35.6,2; 39.7,2; 37.1,2; 39.4,2; 37.3,2; 38.1,2; 38.8,2;
37.6,2; 37.6,2; 39.5,2; 38.1,2; 38.0,2; 37.7,2; 39.7,2; 41.5,2; 37.3,2;
38.4,2; 37.6,2; 38.8,2;
32.3,3; 29.7,3; 30.7,3; 32.0,3; 34.7,3; 29.2,3; 33.5,3; 27.7,3; 34.7,3;
30.2,3; 32.2,3; 31.0,3; 29.8,3; 27.2,3; 32.5,3; 35.1,3]
```

Dataset: 58 patterns

Class 1 = 21

Class 2 = 21

Class 3 = 16

SVM classifier

Err_perc = 16.1111

Err_number = 29

Decision tree classifier

Err_perc = 7.2222

Err_number = 13

Árvore de decisão:

```
inp 1 <= 34.7: out 3 (10.0)
inp 1 > 34.7
| inp 1 <= 39.7: out 2 (10.0)
| inp 1 > 39.7: out 1 (10.0)
```

Testar com a primeira imagem:

Características extraídas:

```
image_0001.jpg: [{"faceRectangle":{"top":206,"left":475,"width":282,"height":282},"faceAttributes":{"age":45.6}}]
```

Vetor:

[45.6,1]

Após correr o classificador o resultado é:

```
pessoa 1
```

- **Apresentação de resultados experimentais, usando imagens dos elementos do grupo de trabalho (ou de outras pessoas à sua escolha).**

- 40602.jpg

Algumas das possíveis características a extrair:

```
[{"faceRectangle":{"top":365,"left":278,"width":530,"height":530},"faceLandmarks":{"pupilLeft":{"x":431.6,"y":496.9},"pupilRight":{"x":665.4,"y":511.1},"noseTip":{"x":548.7,"y":672.4},"mouthLeft":{"x":426.4,"y":742.9},"mouthRight":{"x":641.8,"y":753.2},"eyebrowLeftOuter":{"x":308.6,"y":464.9},"eyebrowLeftInner":{"x":508.3,"y":486.8},"eyeLeftOuter":{"x":388.6,"y":498.4},"eyeLeftTop":{"x":424.8,"y":482.4},"eyeLeftBottom":{"x":419.5,"y":515.1},"eyeLeftInner":{"x":458.8,"y":507.2},"eyebrowRightInner":{"x":596.1,"y":477.8},"eyebrowRightOuter":{"x":766.1,"y":477.1},"eyeRightInner":{"x":618.1,"y":519.6},"eyeRightTop":{"x":659.7,"y":496.0},"eyeRightBottom":{"x":662.8,"y":530.5},"eyeRightOuter":{"x":701.2,"y":516.6},"noseRootLeft":{"x":510.4,"y":523.7},"noseRootRight":{"x":580.8,"y":527.2},"noseLeftAlarTop":{"x":490.1,"y":619.8},"noseRightAlarTop":{"x":595.2,"y":617.5},"noseLeftAlarOutTip":{"x":459.5,"y":657.8},"noseRightAlarOutTip":{"x":657.8,"y":657.8}}}]
```



```
"":{"x":621.0,"y":659.5},"upperLipTop":{"x":540.4,"y":758.4},"upperLipBottom":{"x":540.3,"y":772.5},"underLipTop":{"x":540.2,"y":765.6},"underLipBottom":{"x":540.8,"y":800.6}},"faceAttributes":{"gender":"female","age":23.4}}]sara
```

Características extraídas:

```
40602.jpg: [{"faceRectangle":{"top":365,"left":278,"width":530,"height":530},"faceAttributes":{"age":23.4}}]
```

Vetor:

```
[23.4,1]
```

Após correr o classificador o resultado é:

```
Sara Sobral
```

- 40686.jpg

Algumas das possíveis características a extrair:

```
[{"faceRectangle":{"top":214,"left":89,"width":230,"height":230},"faceLandmarks":{"pupilLeft":{"x":147.6,"y":287.5},"pupilRight":{"x":245.9,"y":268.6},"noseTip":{"x":202.1,"y":334.6},"mouthLeft":{"x":167.6,"y":392.9},"mouthRight":{"x":261.3,"y":373.7},"eyebrowLeftOuter":{"x":98.1,"y":272.8},"eyebrowLeftInner":{"x":162.2,"y":258.4},"eyeLeftOuter":{"x":128.7,"y":292.6},"eyeLeftTop":{"x":144.4,"y":280.6},"eyeLeftBottom":{"x":147.6,"y":295.9},"eyeLeftInner":{"x":164.9,"y":286.7},"eyebrowRightInner":{"x":218.7,"y":248.6},"eyebrowRightOuter":{"x":291.7,"y":240.1},"eyeRightInner":{"x":226.6,"y":276.3},"eyeRightTop":{"x":242.1,"y":262.6},"eyeRightBottom":{"x":247.0,"y":277.6},"eyeRightOuter":{"x":260.6,"y":267.2},"noseRootLeft":{"x":174.7,"y":284.8},"noseRootRight":{"x":212.8,"y":280.3},"noseLeftAlarTop":{"x":176.7,"y":322.2},"noseRightAlarTop":{"x":222.0,"y":315.5},"noseLeftAlarOutTip":{"x":170.5,"y":342.0},"noseRightAlarOutTip":{"x":236.1,"y":331.8},"upperLipTop":{"x":208.7,"y":375.0},"upperLipBottom":{"x":211.3,"y":386.6},"underLipTop":{"x":213.0,"y":387.8},"underLipBottom":{"x":215.9,"y":403.4}},"faceAttributes":{"gender":"male","age":30.2}}]eduardo
```

Características extraídas:

```
40686.jpg: [{"faceRectangle":{"top":214,"left":89,"width":230,"height":230},"faceAttributes":{"age":30.2}}]
```

Vetor:

```
[30.2,2]
```

Após correr o classificador o resultado é:

```
Eduardo António
```