

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES
MESTRADO EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES
PROCESSAMENTO DE IMAGEM E BIOMETRIA

Trabalho Prático 2

Semestre de verão 2016/2017 (22 de maio de 2017)

Data Limite de Entrega (Código e Relatório): 10 de julho de 2017

Objetivos:

- Realização de processamento digital de imagem, sobre imagens coloridas, monocromáticas e binárias.
 - Resolução de problemas comuns de processamento digital de imagem, com ênfase na área da biometria.
 - Desenvolvimento e avaliação de um sistema biométrico de deteção e reconhecimento facial.
-

O código desenvolvido e o respetivo relatório deverão ser entregues em formato eletrónico no sistema Thoth, até 10 de julho de 2017.

1. Dada a função `medical_image_enhancement.m`, desenvolvida no exercício 2i) do primeiro trabalho prático, bem como a implementação da coloração de imagem, desenvolvida no exercício 5 do primeiro trabalho prático:
 - a) Realize a coloração das imagens médicas originais e das imagens produzidas pela função `medical_image_enhancement.m`, recorrendo a uma técnica de coloração à sua escolha.
 - b) Considere as imagens médicas do conjunto `MedicalImages.zip`. Apresente todas as imagens coloridas (antes e após a aplicação da função `medical_image_enhancement.m`).
 - c) Comente os resultados obtidos na alínea anterior. Existe diferença apreciável entre os resultados da coloração da imagem original e da coloração da imagem produzida pela função `medical_image_enhancement.m`?
2. Tenha em conta os conceitos de *web safe color* (216 cores) e *web safest color* (22 cores), tal como apresentando na página WEB https://en.wikipedia.org/wiki/Web_colors.
 - a) Escreva a função/método `rgb2safeColor`, a qual converte uma imagem RGB na sua versão *web safe color* ou *web safest color*, mediante um parâmetro de entrada. A função/método deverá indicar ou retornar o número de cores distintas da imagem resultante. Indique o critério aplicado para determinar as cores *safe* e *safest*.
 - b) Aplique a função/método `rgb2safeColor` a duas imagens produzidas pela função/método `codeCardGenerator`, do exercício 4 do primeiro trabalho prático. Para as duas imagens, compare a versão original, a versão *safe* e a versão *safest*. Comente relativamente à qualidade visual de cada imagem.
3. Pretende-se melhorar o desempenho da função `fingerprint_enhancement.m`, desenvolvida no exercício 2ii) do primeiro trabalho prático, através da introdução de processamento morfológico. Em seguida, pretende-se processar as imagens resultantes, para a deteção de minúcias.
 - a) Escreva a função `fingerprint_enhancement_morph.m`, a qual melhora a função anterior, através da aplicação de operações morfológicas adequadas.
 - b) Apresente os resultados experimentais obtidos com as imagens do conjunto `FingerprintImages.zip`, pelas funções `fingerprint_enhancement.m` e `fingerprint_enhancement_morph.m`. Compare e comente os resultados.
 - c) Escreva a função `minutiae_detection.m`, a qual realiza a deteção e a marcação de minúcias sobre a imagem binária de entrada. A função deverá afixar, sobre a imagem de entrada, todos os pontos detetados. Apresente os resultados obtidos e comente sobre a qualidade dos mesmos.
4. Projete, realize e avalie um módulo de deteção e reconhecimento facial, para um número reduzido de utilizadores (três ou quatro).
 - a) Descreva como realiza as funcionalidades de deteção e de reconhecimento.
 - b) Apresente resultados experimentais de deteção e reconhecimento, usando imagens de uma base de dados específica para este efeito ou do conjunto `FaceImages.zip`.
 - c) *Opcional. Apresente resultados experimentais, usando imagens dos três elementos do grupo de trabalho (ou de outras pessoas à sua escolha).*

De seguida, apresentam-se alguns recursos sugeridos para a elaboração deste trabalho.

Bibliografia (exercícios 3 e 4):

- R. Leitão, Sistema de Reconhecimento de Impressões Digitais Baseado em FPGA, dissertação de mestrado, ISEL, 2014, disponível em <http://repositorio.ipl.pt/bitstream/10400.21/4199/1/Dissertação.pdf>
- N. Zaeri, "Minutiae-based Fingerprint Extraction and Recognition", InTech Open, online, at <https://cdn.intechopen.com/pdfs-wm/16502.pdf>
- P. Parra, "Fingerprint minutiae extraction and matching for identification procedure", University of California, San Diego, online, at <https://cseweb.ucsd.edu/classes/wi07/cse190-a/reports/pparra.pdf>
- D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, "Handbook of Fingerprint Recognition", Springer-Verlag New York, Inc., 2003.
- S. Li and A. Jain, "Handbook of Face Recognition", Springer Publishing Company, Inc., 2nd edition, 2011.

Bases de dados de imagens de face (exercício 4):

- **The Yale Face Database**, <http://vision.ucsd.edu/content/yale-face-database>.
- **Caltech Face Database**, <http://www.vision.caltech.edu/html-files/archive.html>.
- **Georgia Tech Face Database**, http://www.anefian.com/research/face_reco.htm.
- **The AR Face Database**, <http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html>.

Software (exercícios 2 e 4):

- **OpenCV**, <http://opencv.org/>, an open source computer vision and machine learning software library.
- **ImageJ**, <http://imagej.nih.gov/ij/index.html>, Image Processing and Analysis in Java.
- **JVT - Java Vision Toolkit**, <http://javavision.sourceforge.net/>.
- **Java DIP - Digital Image Processing**, http://www.tutorialspoint.com/java_dip/index.htm
- **AForge.NET**, <http://www.aforge.net.com/framework/>, Open Source C# Framework.
- **Weka**, <http://www.cs.waikato.ac.nz/ml/weka/>, a collection of machine learning algorithms for data mining tasks.