FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Advanced Topics in Digital Image Processing

LESSON 0 - *INTRODUCTION*

Objective:

Install the software and libraries and test some basic image processing functions.

Procedure:

Preparation

1. Install Python environment version 3.8 and PyCharm Community Edition IDE

   Python 3.8.10 (windows)

   https://www.python.org/downloads/release/python-3810/

   PyCharm:

   https://www.jetbrains.com/pycharm/download/#section=windows

2. Prepare Libraries: On PyCharm's python command prompt, run the following lines to install these libraries:

```
pip install numpy
pip install queues
pip install opencv-python
pip install matplotlib
```

Note: if *pip* is not installed, you can follow the instructions on:

https://www.geeksforgeeks.org/how-to-install-pip-on-windows/

3. Create a python script file to open and show an image

```
import cv2 as cv

#abrir imagem
pathname = "..\\Aula 0\\images\\"
filename = "peppers.jpg"

img = cv.imread(pathname + filename)

cv.imshow("Imagem", img)
```

4. Add a file chooser window to get the filename

```
import tkinter as tk

root = tk.Tk()
root.withdraw()

file_path = tk.filedialog.askopenfilename()
```

5. Change the image colors to grayscale using the function :

```
imgGray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
```

6. Apply an image Threshold to get a black & white image:

```
ret,imgBW = cv.threshold(imgGray,12,255, cv.THRESH_BINARY )
```

7. Add “ `| cv.THRESH_OTSU`” after `cv.THRESH_BINARY` and comment the result

8. Add the imageForms.py python file to the project and test the *showSideBySideImages* function

9. Apply a blur filter with a window size (ksize) of 3x3 using the *cv.blur* function

10. Accessing each pixel’s values (with a *for* cicle), adjust the image brightness and contrast according with the user input (python function *input*).

    pixel_new = pixel * contrast + brightness

    hints: use image’s shape to get the dimensions and get pixel value using img[x, y, c]