Erick de Sousa Almeida - Bi3008444

Atividades - Segmentação Parte 1

1.

Implementar limiarização, definir

```
import os

import cv2
import numpy as np
```

A limiarização é uma técnica de segmentação de imagens que consiste em separar os píxeis de uma imagem em dois grupos: pixels que possuem intensidade maior que um limiar pré-definido e píxeis que possuem intensidade menor que o limiar.

```
def thresholding(img, limiar):
    img_out = np.zeros(img.shape)
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            if img[i, j] > limiar:
                img_out[i, j] = 255
            else:
                img_out[i, j] = 0
    return img_out
```

2.

Implementar detector de bordas Canny, e aplica o filtro de borramento(filtro gaussiano) e verificar se o borramento melhora a detecção de bordas.

O detector de bordas Canny é um dos mais utilizados na literatura. Ele é composto por 4 etapas:

1. Suavização da imagem com um filtro gaussiano
2. Cálculo do gradiente da imagem
3. Supressão de não-máximos
4. Limiarização com histerese

Seguida da detecção das bordas.

```
def canny(img, sigma=0.33, t1=0.1, t2=0.2):
    # 1. Suavização da imagem com um filtro gaussiano
    img = gaussian_blur(img, 5)
    # 2. Cálculo do gradiente da imagem
    img = cv2.Canny(img, int(t1 * 255), int(t2 * 255))
    return img

def gaussian_blur(img, kernel_size=5):
    return cv2.GaussianBlur(img, (kernel_size, kernel_size), 0)
```

2.1

Mudar os parâmetros T1 e T2 e avaliar a qualidade das bordas detectadas.

```
params = [(0.1, 0.2), (0.2, 0.3), (0.3, 0.4), (0.4, 0.5), (0.5, 0.6)]
images_names = os.listdir('images')
images = [cv2.imread('images/' + name, 0) for name in images_names]
images = [canny(img, t1=t1, t2=t2) for img in images for t1, t2 in
params]

for i in range(len(images)):
    cv2.imwrite('results/' + images_names[i // len(params)] + '_' +
str(i % len(params)) + '.png', images[i])
```