CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

Problem Set 2 Report

Problem Set 2 from Computer Vision Course Proposed by professor Flávio Cardeal

Ramon Griffo Costa

May 2019
Belo Horizonte

Introduction

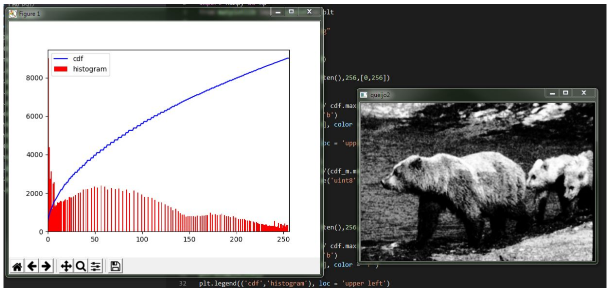
All the codes were implemented in Python 3.6 using the libraries opencv, numpy and matplotlib, the codes and images used are attached together with this report.

1)

- The code implemented loads an image and applies a blur followed by an histogram equalization, defined by the following formula:

$$g^{(r)}_{equal}(u) = \frac{G_{\max}}{Q}\sum_{w=0}^{u} h_I(w)^r \quad \text{with } Q = \sum_{w=0}^{G_{\max}} h_I(w)^r$$

- When the value of R is equal to 1 the cumulative distribution function tends to linear and the image becomes less noisy. (Image 3)
- When the value is greater than 1 the image becomes darker and the cumulative distribution function has a decreasing slope, and the image looks even less noisy. The values look more equalized. (Image 1)
- When the value is lesser than 1 the image looks brighter and the cumulative distribution function slope is increasing, and the image looks overexposed and not so clean. The values look less equalized. (Image 2)
- And for r=0 the image is all black, because the cumulative value has to be constant, so all pixels have to be black. (Image 4)



```
plt.legend(('cdf','histogram'), loc = 'upper left')
```

Image 1 - R>1

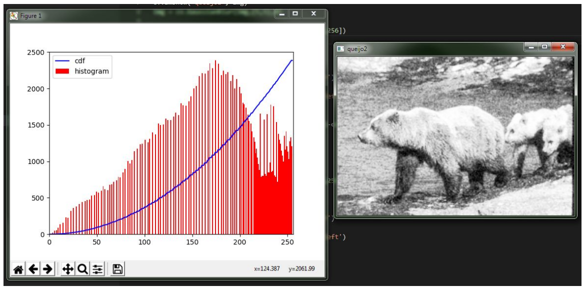


Image 2 - R<1

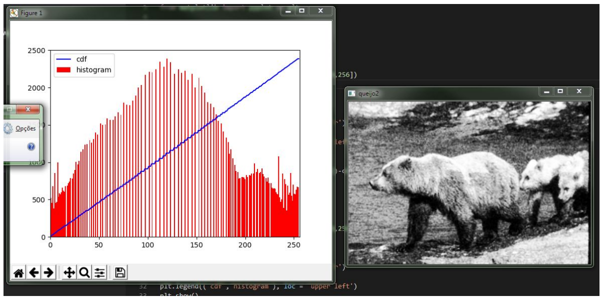


Image 3 - R=1

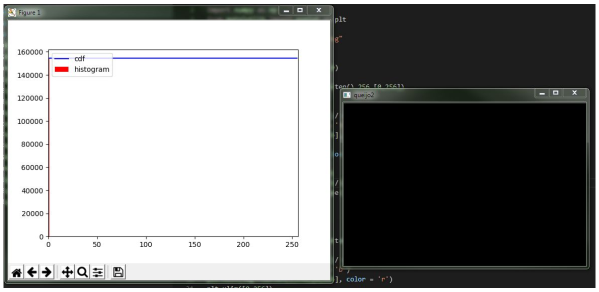
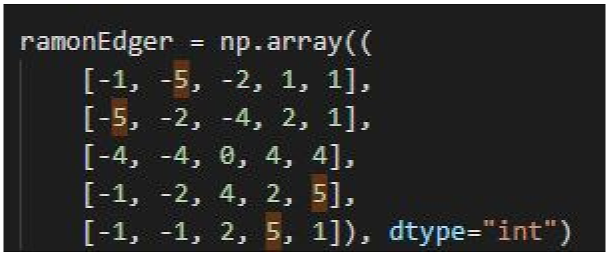


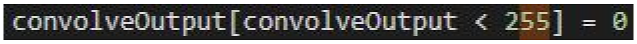
Image 4 - R=0

2)

- The code implements a convolution operation, and a specific convolution matrix to detect edges, the matrix was based in the idea of sobel operators, but in a larger matrix (5x5) and thinking in both x and y directions and was changed experimentally until a reasonable result was found, then due to lots of noise in the result image the only values shown are the ones equal to 255, the final results look quite good.



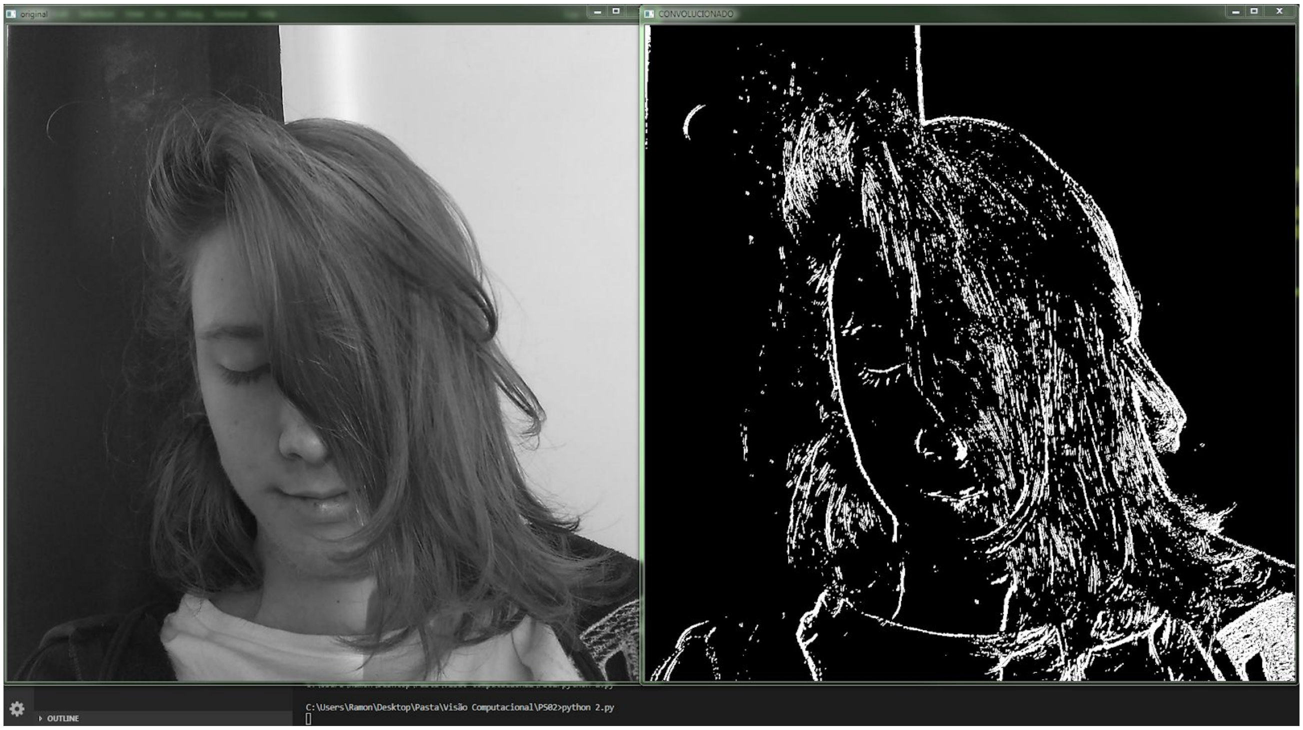
```
ramonEdger = np.array((
    [-1, -5, -2, 1, 1],
    [-5, -2, -4, 2, 1],
    [-4, -4, 0, 4, 4],
    [-1, -2, 4, 2, 5],
    [-1, -1, 2, 5, 1]), dtype="int")
```



```
convolveOutput[convolveOutput < 255] = 0
```

Image 5 - Convolution Matrix

Image 6 - Dropping all values less than 255



```
C:\Users\Ramon\Desktop\Pasta\Visão Computacional\PS02>python 2.py
```

Image 7 - Results

3)