
Relatório - Atividade 0

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE COMPUTAÇÃO

INTRODUÇÃO AO PROCESSAMENTO DIGITAL DE IMAGENS

MARÇO DE 2018

FELIPPE TRIGUEIRO ANGELO
210479

1 Descrição geral da atividade 0

A primeira atividade da disciplina Introdução ao Processamento de Imagens teve como objetivo, o alinhamento de algumas ferramentas que serão necessárias no decorrer do semestre, e.g. as bibliotecas *numpy*, *matplotlib*, *scipy*, entre outras. Para atingir tal, foi pedida a implementação de um programa que extraísse algumas características das imagens, bem como realizasse algumas transformações nas mesmas.

2 Executando o Programa

Para executar o programa, entre no terminal do linux, navegue até a pasta onde os códigos estão localizados e digite o seguinte comando:

```
sudo python3 atividade0.py [nome-das-imagens]  
Ex: sudo python3 atividade0.py city.png baboo.png seagull.bmp
```

2.1 Dados de Entrada

Para a correta execução do programa é necessário que as imagens de entrada estejam na mesma pasta que os códigos fonte.

2.2 Dados de Saída

Após o processamento e extração de algumas características, o programa gera como saída um conjunto de duas imagens para cada imagem de entrada, que correspondem ao negativo e a mudança de escala da respectiva imagem de entrada. As imagens de saída estão localizadas na pasta /Output_Images e estão no formato .png. Também são extraídas informações como dimensões, menor, maior e intensidade média, que são posteriormente exibidas no terminal.

3 Descrevendo o Código

3.1 Algoritmos e Estruturas de Dados utilizadas

3.1.1 Conversão de Escala

Uma das atividades requeridas foi o mapeamento dos valores de intensidades, que estavam na escala $[0; 255]$, para a escala $[120; 180]$. Para tal, foi desenvolvida a função `imconv(*)` que recebe como parâmetros de entrada, a imagem na escala que se deseja converter, a escala de entrada e de saída. Como retorno a função devolve uma imagem X_o na escala desejada.

A conversão foi feita por meio da igualdade das proporções:

$$\frac{X_i - 0}{X_o - 120} = \frac{255 - 0}{180 - 120}$$

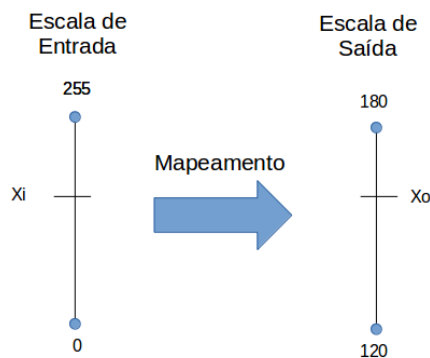


Figure 1: Mapeamento de uma escala em outra.

improc.py

```
1 import numpy as np
2
3 def imconv (Xi, vmin_i, vmax_i, vmin_o, vmax_o):
4     Xo = (Xi - vmin_i)*((vmax_o - vmin_o)/(vmax_i - vmin_i)) + vmin_o
5     Xo = Xo.astype(int)
6
7     return Xo
```

3.1.2 Negativo da imagem

Calcular o negativo de uma imagem consiste em um caso particular de conversão de escala, onde a escala de saída é o inverso da escala de entrada. Assim, seguindo a lógica da conversão anterior, tem-se o seguinte arranjo:

$$\frac{X_i - 0}{X_o - 255} = \frac{255 - 0}{0 - 255},$$

portanto encontra-se a relação para o negativo da imagem:

$$X_o = 255 - X_i$$

```
1 #IMAGEM NEGATIVA
2 img_neg = 255-img          #Calcula o negativo da imagem
```

3.1.3 Estruturas de dados utilizadas

No código, foram utilizadas as estruturas de dados array e tupla no armazenamento das informações desejadas. No caso das imagens foram utilizados os arrays providos pela biblioteca *numpy*, assim cada imagem pode ser vista como uma matriz bidimensional. Já no armazenamento do histograma, a função *numpy.histogram(*)* retorna uma tupla contendo o valor do histograma e os bins correspondentes. A tupla também é utilizada no armazenamento das dimensões das imagens, por meio da extração do atributo *.shape*.

3.2 Considerações adotadas na solução dos problemas

Na extração das informações de menor, maior e intensidade média dos níveis de cinza, foram utilizadas as respectivas funções da biblioteca *numpy*, *numpy.min()*, *numpy.max()*, *numpy.mean()*. Todas elas operam sobre os arrays da biblioteca *numpy*.

Um outro ponto que convém citar aqui, é que o histograma foi exibido no formato normalizado, com a finalidade de facilitar a comparação entre histogramas de diferentes imagens.

3.3 Testes adotados e Limitações do programa

Foram feitos testes com imagens combinando as características dos grupos a seguir: (i) Imagens Coloridas e Monocromáticas, (ii) Imagens Retangulares e Quadradas, (iii) Imagens localizadas no mesmo e em um diretório diferente, e (iv) Imagens no formato .png, .jpg e .bmp.

Com exceção das imagens localizadas em um diretório diferente daquele que o código está, todas as combinações anteriores resultaram em um êxito na execução. Assim, para facilidade na implementação do código, foi evitado o caso em que as imagens estão em um diretório diferente daquele do código fonte, gerando portanto, uma limitação na forma de se passar uma imagem.

Convém citar aqui, que para imagens coloridas, o histograma corresponde a contagem do nível de iluminação considerando juntamente os três canais do sistema RGB.

Um resultado interessante dos casos testados, foi que para uma mesma imagem representada em formatos diferentes, e.g. png, jpg, bmp, foram obtidos histogramas diferentes.