

Introdução ao processamento digital de imagem – Mestrado

Stephane de Freitas Schwarz

Prof. Dr. Hélio Pedrini

1. Objetivo

O objetivo deste trabalho é realizar alguns processamentos básicos em imagens digitais com o auxílio de algumas bibliotecas próprias para manipulação de imagens e vetores.

2. Requisitos

O algoritmo foi construído para ser executado em qualquer computador desde que atenda as seguintes observações: Python versão 2.7.12 ou superior, bibliotecas scikit-image, numpy, matplotlib e numpy.

3. Executar o programa

Para executar o programa o usuário deve digitar no terminal `python [nome_do_arquivo.py]` pressionar a tecla [Enter] depois `[nome_ou_caminho_da_imagem.png]`.

4. Transformação de cores

Converter uma imagem representada no sistema de cores RGB para tons de cinza hoje é uma tarefa que se restringe na execução de apenas uma linha de código. Na biblioteca scikit-image, por exemplo, a função `rgb2gray` computa a iluminância de uma imagem rgb para tons de cinza multiplicando cada faixa de cor (R,G ou B) por uma constante de calibração. A equação 1 ilustra este processo.

$$I = R * 0,2125 + G * 0,7154 + B * 0,0721$$

Dessa forma o resultado obtido após a execução do comando é a transformação de uma imagem RGB para tons de cinza como mostra a figura 1.

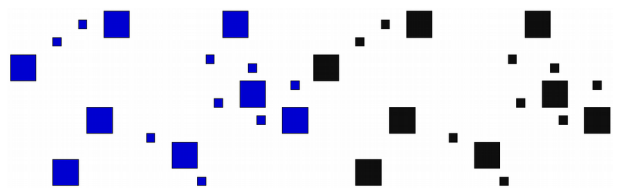


Figura 1 – Transformação de cores.

5. Contorno dos objetos

Embora seja comum associarmos o conceito de bordas e contornos em um mesmo contexto, tais propriedades apresentam diferenças significativas. Detecção de bordas consiste no processo de identificar pontos onde a intensidade do pixel muda drasticamente e atribuir valor lógico alto para estes, enquanto o resto recebe valor lógico baixo. Já os contornos não são necessariamente parte da imagem, mas correspondem a forma dos objetos contidos nela.

Ressaltar o contorno de uma imagem não é uma tarefa trivial, no entanto, com o avanço dos pacotes de processamento de imagem podemos identificá-los mediante a utilização de apenas um comando, que retorna uma lista com as coordenadas exatas do contorno de cada objeto na imagem. A figura 2 mostra a imagem analisada e seus contornos identificados sobrepostos a ela.

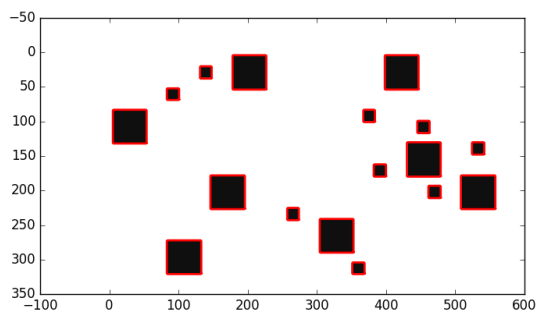


Figura 2 – Contorno de objetos.

6.Extração de propriedades dos objetos

É possível mensurar as propriedades de uma imagem a partir de uma imagem com regiões conectadas, isso é, quando pixels vizinhos contêm o mesmo valor de intensidade.

No pacote Skimage a função responsável por capturar essas propriedades é chamada *regionprops*. O resultado obtido é semelhante a figura 3.

regiao:	0	perimetro:	190	area:	2352
regiao:	1	perimetro:	190	area:	2352
regiao:	2	perimetro:	62	area:	272
regiao:	3	perimetro:	62	area:	272
regiao:	4	perimetro:	188	area:	2304
regiao:	5	perimetro:	62	area:	272
regiao:	6	perimetro:	64	area:	289
regiao:	7	perimetro:	190	area:	2352
regiao:	8	perimetro:	64	area:	289
regiao:	9	perimetro:	64	area:	289
regiao:	10	perimetro:	190	area:	2352
regiao:	11	perimetro:	190	area:	2352
regiao:	12	perimetro:	64	area:	289
regiao:	13	perimetro:	62	area:	272
regiao:	14	perimetro:	188	area:	2304
regiao:	15	perimetro:	190	area:	2352
regiao:	16	perimetro:	62	area:	272

Figura 3 – Propriedades da imagem.

7.Histograma de área dos objetos

Uma vez que são obtidas as propriedades da imagem é possível realizar uma série de manipulações com esses valores. Mostrar cada região rotulada individualmente, verificar se os objetos são grandes, pequenos ou médios segundo um determinado critério são apenas alguns exemplos do que se pode fazer com tais medidas. As figuras 4, 5 e 6 ilustram os exemplos supramencionados.

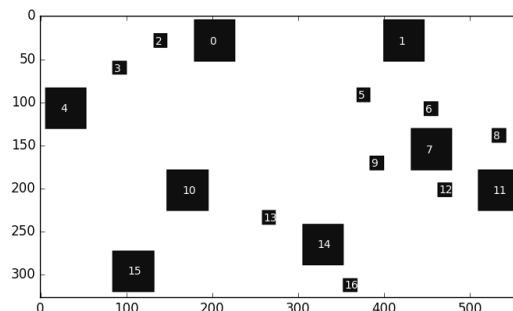


Figura 4 – Imagem rotulada.

```
numero de regioes pequenas 9
numero de regioes medias 8
numero de regioes grandes 0
```

Figura 5 – Classificação de medidas.

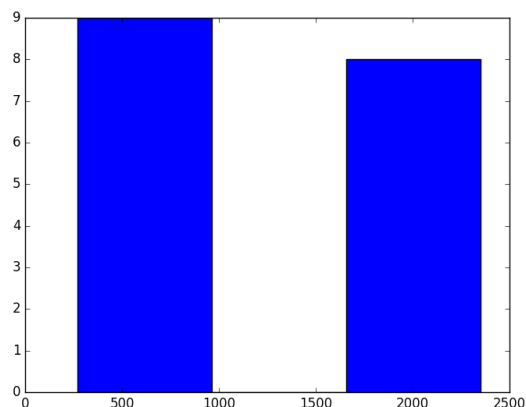


Figura 6 – Histograma da imagem.