Trabalho 4 - MC920

Rian Radeck Santos Costa - 187793

10 de Novembro de 2022

Sumário

1 Transformação de Cores

O objetivo dessa parte é binarizar a imagem para separarmos fundo de objeto. Para realizar esta etapa utilizei de uma função do openCV que converte uma imagem colorida para tons de cinza (cv2.cvtColor) e usei a função de threshold do openCV (cv2.threshold) para binarizar a imagem, de tal forma que onde fosse branco seria fundo e qualquer pixel diferente de branco seria objeto. Essa parte do código está em uma função com o nome "monochromatic" que recebe uma imagem colorida e retorna a imagem monocromática.

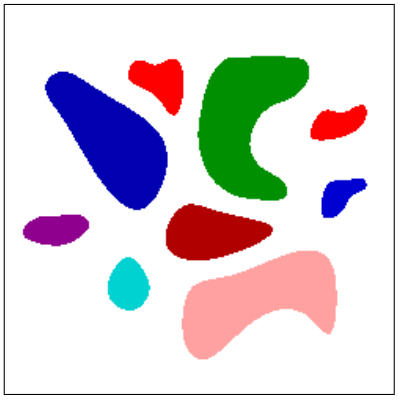


Imagem original

Imagem monocromática

2 Contornos dos Objetos

O algoritmo aplicado para resolver esta etapa foi o seguinte:

- Itere por todos os pixeis da imagem.
- Para cada pixel veja se algum vizinho dele (*4-neighbourhood*) é fundo.
- Se o item anterior for verdade esse pixel é borda.

No código decidi colorir as bordas de azul e a implementação está na função "borders", que recebe uma imagem, colorida ou em tons de cinza, e retorna uma imagem somente com as bordas dos objetos da imagem recebida. Veja o exemplo abaixo:

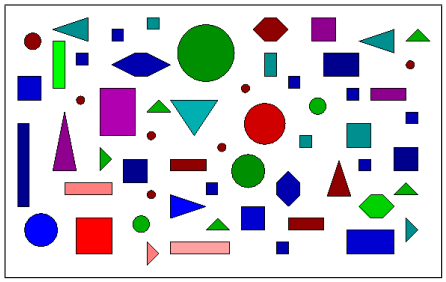
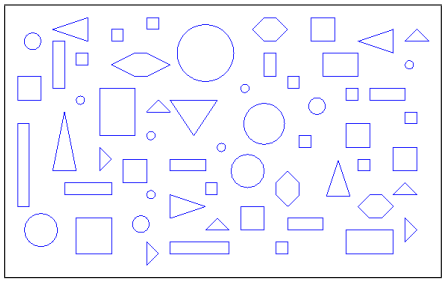


Imagem original



Bordas da imagem

3 Extração de Propriedades dos Objetos

Para extrair os dados dos objetos, separei eles em componentes conexas e utilizei a biblioteca "scikit-image".

O algoritmo usado para a separação das componentes foi o seguinte:

1. Para cada pixel da imagem verifique:
   - Se é objeto.
   - Se não pertence a nenhuma componente no momento.
2. Se ambos são verdade inicie um algoritmo de flood fill a partir desse pixel. (Utilizei BFS)

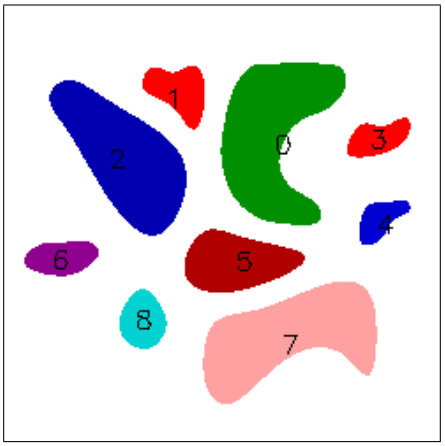
Na BFS guardei: ID da componente, menor e maior linha atingidas na busca, menor e maior coluna atingidas na busca, coordenadas de todos os pixeis da componente.

Para gerar as imagens de cada objeto fiz para cada uma das componentes achadas na BFS, um slicing da imagem original de acordo com as linhas e colunas salvas e recolori todos os pixeis desse slicing de acordo com se ele estava ou não na lista de coordenadas salvas daquela componente.

Tendo essas imagens apenas apliquei a função measure.regionprops da biblioteca citada anteriormente. O resultado detalhado para cada imagem está na pasta processed deste trabalho.

Para rotular a imagem foi bem simples uma vez que tinha salvo a borda dos objetos na BFS. Utilizei a função cv.putText para escrever a ID da componente que representa aquele objeto e centralizei o texto com o centro do objeto.

Imagem original



Regiões rotuladas

Região	Área	Perímetro	Excentricidade	Solidez
0	3969	313,76	0,82	0,75
1	791	119,98	0,74	0,90
2	3584	259,46	0,90	0,98
3	540	99,25	0,89	0,90
4	438	88,77	0,86	0,91
5	1684	174,12	0,87	0,97
6	642	103,01	0,89	0,96
7	3934	305,42	0,91	0,77
8	675	96,33	0,62	0,97

Propriedades dos objetos

4 Histograma de Área dos Objetos

Sem muita complicação, o único detalhe aqui é que a última marcação no eixo x é o máximo entre 4500 e a maior área dos objetos (observe no exemplo 2).

4.1 Exemplo 1

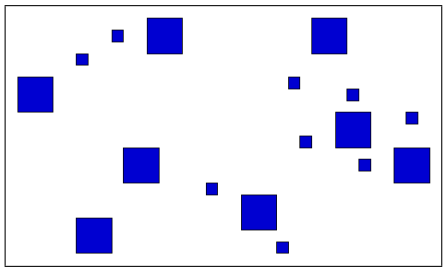
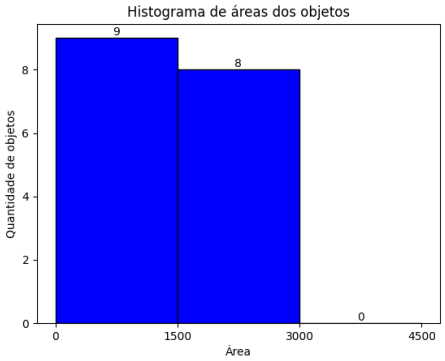


Imagem original



Histograma

4.2 Exemplo 2

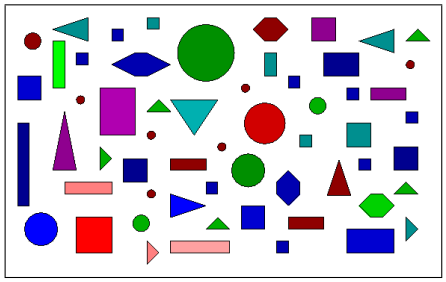
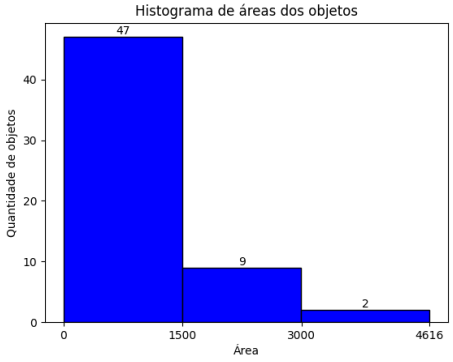


Imagem original



Histograma

4.3 Exemplo 3

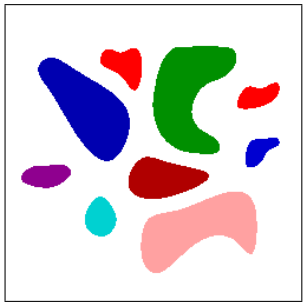
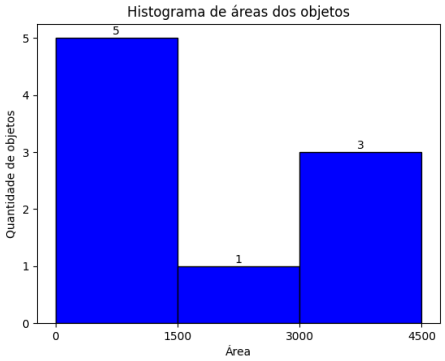


Imagem original



Histograma

5 Bibliografia

Documentação OpenCV
Documentação Matplotlib
Documentação Scikit-Image