Relatório Trabalho 1

MC920 – Introdução ao Processamento de Imagem Digital

Julianny Favinha Donda RA 156059

1. Solução

1.1. Transformação de cores

Sabemos que a imagem de entrada é formada pelo conjunto de cores RGB, onde o fundo é formado apenas por *pixels* brancos. Para converter a imagem para níveis de cinza, a função utilizada foi `skimage.color.rgb2gray()`, que converte todos os *pixels* para o intervalo [0, 1]. Como queremos uma imagem monocromática na escala [0, 255] e sabemos que o fundo possui apenas *pixels* brancos, fazemos então outra transformação de tal forma que os *pixels* que possuem valor menor que 255 fazem parte dos objetos e, consequentemente, terão valor 0. Podemos transformar a imagem para monocromática pois há a garantia de que nenhum dos objetos estão sobrepostos. A imagem foi salva usando as funções `matplotlib.pyplot.imshow()` e `matplotlib.pyplot.savefig()`.

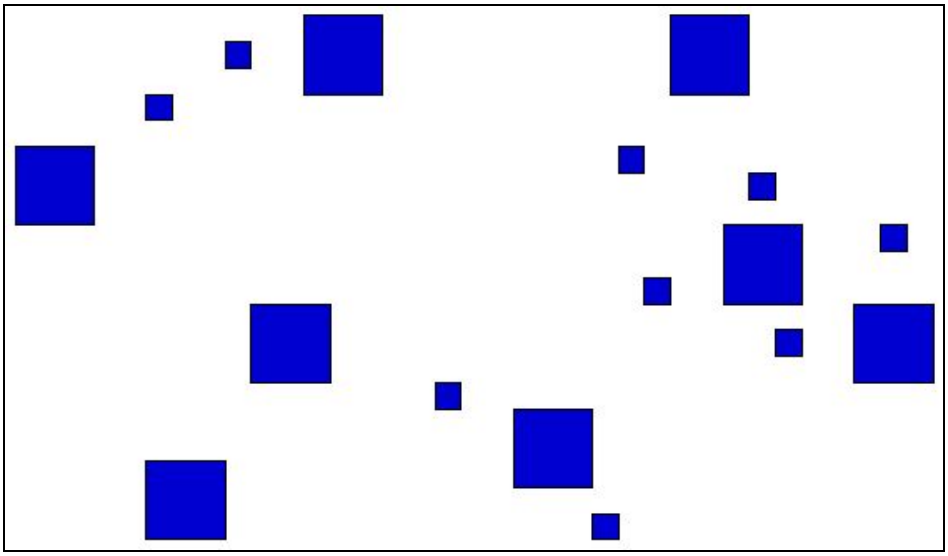

Figura 1 - Imagem original, de nome `objetos1.png`.

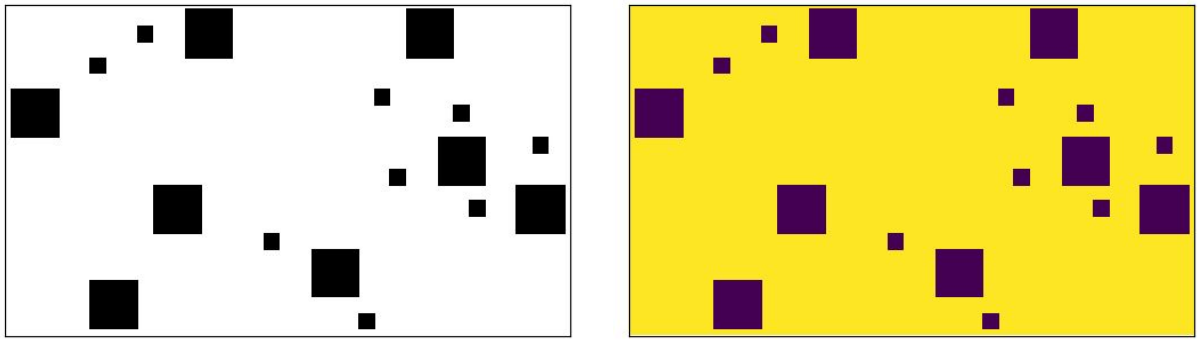

Figura 2 - Função `imshow` da biblioteca `matplotlib` necessita do parâmetro `cmap="gray"` para exibir uma imagem em escala de cinza (esquerda). Caso contrário, a função utiliza um mapa de calor para exibir a imagem (direita).

1.2. Contornos dos objetos

Para encontrar as bordas dos objetos presentes na imagem, foi usado a função `skimage.measure.find_contours()`.

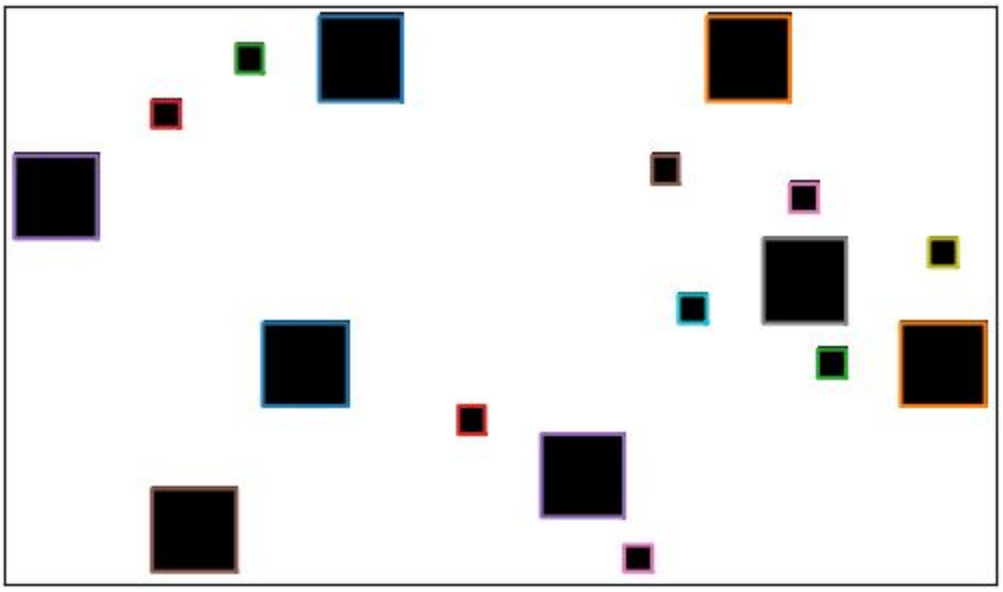

Figura 3 - Imagem obtida após executar a função `find_contours()`. Note que há objetos onde o contorno não é exato, o que indica que a função utilizada possui um erro de precisão..

1.3. Extração de propriedades dos objetos

As propriedades dos objetos da imagem foram obtidas através das funções `skimage.measure.label()` e `skimage.measure.regionprops()`. A primeira função rotula as regiões e, nesse caso, foi escolhido o critério de vizinhança-4. A partir da imagem com *labels*, a segunda função calcula as propriedades de cada região. Para este trabalho, estamos interessados nas propriedades de *label*, *perimeter* e *area* e estas propriedades são exibidas na saída padrão.

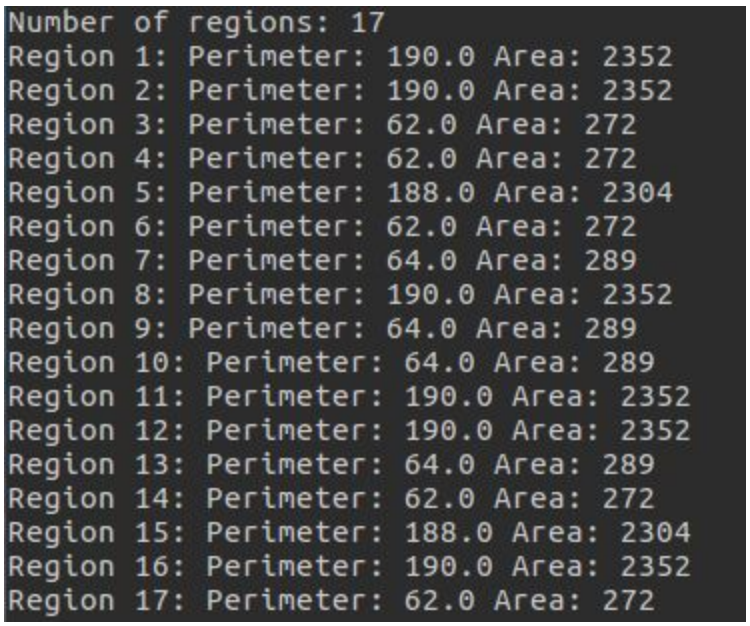

Figura 4 - Captura de tela da saída do programa usando a imagem `objetos1.png` e destacando as propriedades exibidas. A propriedade *label* é o número dado para cada região, descrito em "Region".

1.3.1. Classificação de objetos de acordo com a área

Para cada área encontrada (Figura 4), as regiões (ou objetos) foram classificadas da seguinte maneira:

Classificação	Intervalo
Pequeno	Área < 1500 pixels
Médio	Área ≥ 1500 pixels e Área < 3000 pixels
Grande	Área > 3000 pixels

Tabela 1 - Classificação de objetos a partir do valor de sua área.

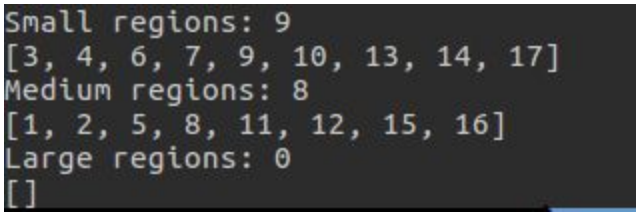

Figura 5 - Captura de tela da saída do programa usando a imagem `objetos1.png` e destacando as regiões classificadas de acordo com o critério da Tabela 1.

1.3.2. Centróide

Além das propriedades listadas acima, também estamos interessados na propriedade *centroid*, que é o ponto central da região encontrada. Usamos essa informação para escrever o *label* de cada região na imagem. Para escrever um texto na imagem, usamos a função `matplotlib.pyplot.text()`.

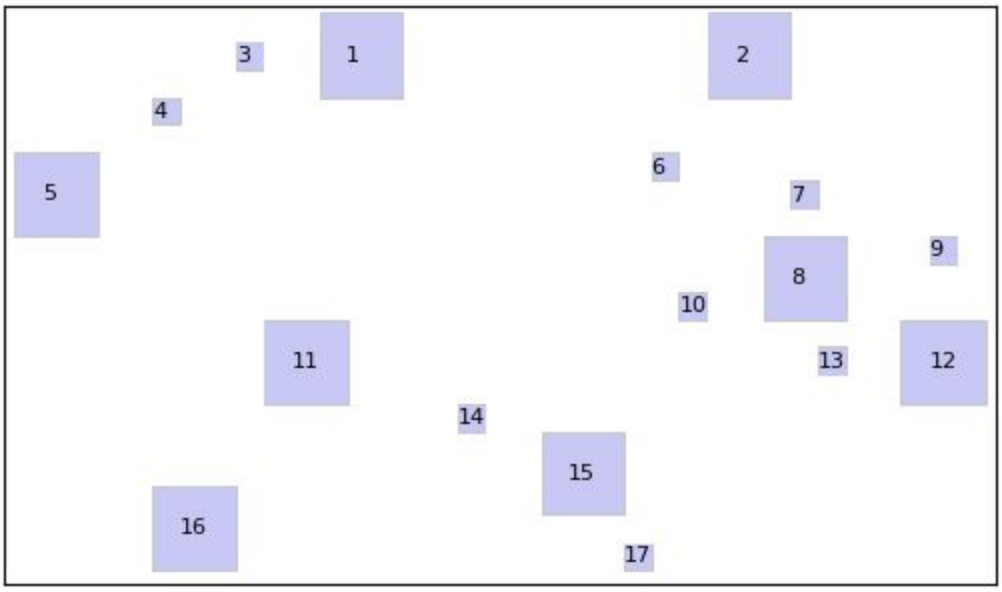

Figura 6 - Imagem destacando os *labels*, que foram colocados próximos ao centróides de cada região. Para fins de visibilidade, a imagem foi transformada para outra escala de intensidades, usando a função `skimage.exposure.rescale_intensity()`.

1.4. Histograma de áreas dos objetos

O histograma foi obtido através da função `matplotlib.pyplot.hist()`, utilizando um *array* das áreas obtidas anteriormente.

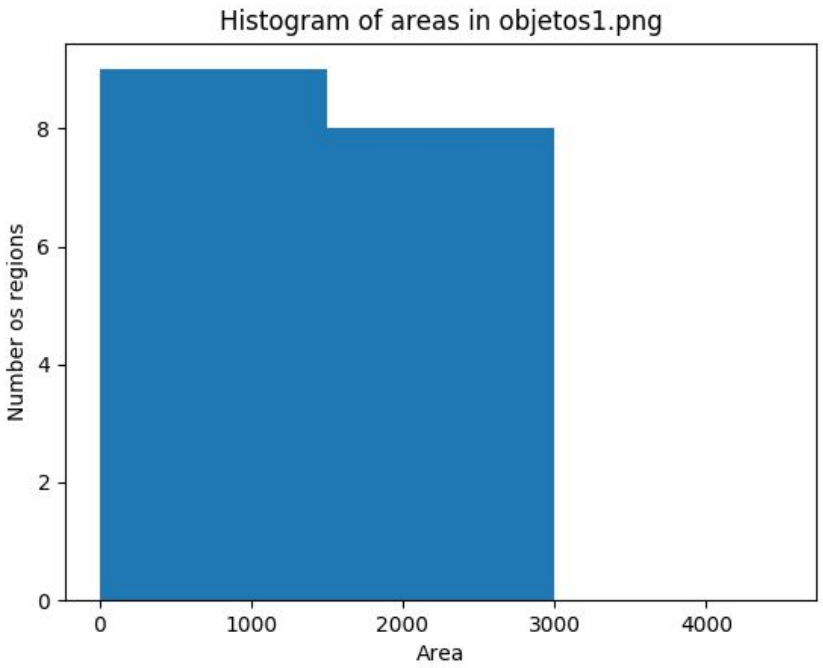

Figura 7 - Histograma das áreas das regiões obtidas da imagem `objetos1.png`.

Note que o eixo x (área) do histograma foi dividido de tal forma que é fácil ver que no intervalo $[0, 1500]$ temos 9 regiões, no intervalo $[1500, 3000]$ temos 8 regiões e no intervalo $[3000, 4500]$ temos 0 regiões. Isso conforma com o resultado obtido na Figura 5.

2. Execução do código

É necessário que as bibliotecas **numpy**, **matplotlib** e **scikit-image** estejam instaladas. No terminal, execute

```
$ python3 lab1.py [nome da imagem].png
```

O programa `lab1.py` gera três arquivos

- [nome da imagem]_blackandwhite.png
- [nome da imagem]_contours.png
- [nome da imagem]_histogram.png
- [nome da imagem]_labelled.png

Que indicam, respectivamente, a imagem monocromática, a imagem com os contornos dos objetos, o histograma das áreas dos objetos e a imagem com rótulos em cada objeto. Além disso, é possível visualizar outras informações na saída padrão, como descrito nas figuras 4 e 5.