

Relatório Processamento de Imagens 1

RA: 172017 - Leonardo Maldonado Pagnez

28 de Março de 2018

1 Introdução

Durante as últimas semanas da disciplina MC920 - Introdução ao Processamento de Imagem Digital, conceitos de realce, construção de histograma para imagens e vizinhança foram introduzidos. Para a aplicação prática dos conceitos teóricos, o trabalho 1 foi uma maneira de trazer estes termos e suas implicações para fora da aula.

Os problemas deste trabalho podem e foram organizados em itens de 1 a 4, sendo eles: 1.1 A transformação de cores da imagem; 1.2 Traçar os contornos da imagem; 1.3 Extração das propriedades de área, perímetro e centroide da imagem; 1.4 Classificação das áreas dos Objetos expostos em um histograma. As principais entradas utilizadas para resolver o problema foram extraídas do website da disciplina, oferecidas pelo professor responsável pela disciplinas, e exibidas a seguir:

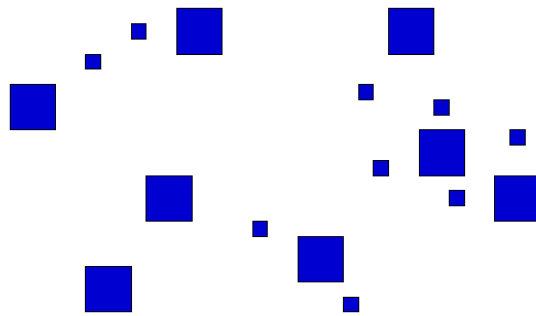


Figure 1: Imagem 'objetos1.png' de entrada para o programa

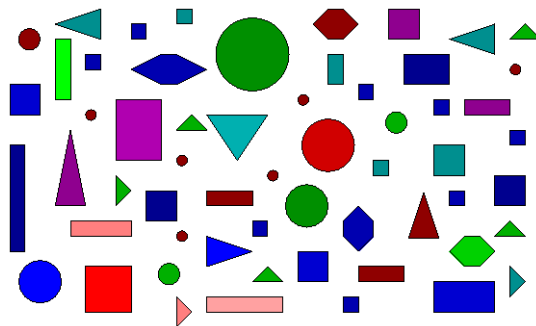


Figure 2: Imagem 'objetos2.png' de entrada para o programa

Como já mencionado pelo professor responsável, a disciplina não foi feita para colocar alunos na posição de desenvolvedores de cada função de alteração de contorno, cálculo de áreas de pixels, entre outros; logo foi fundamental o uso de bibliotecas que conseguem reproduzir as técnicas escolhidas pelo aluno para resolver

os problemas propostos pelo enunciado deste trabalho 1. As bibliotecas escolhidas pelo aluno durante esse trabalho foram em destaque: **skimage** - com suas excelentes funções de rotulação de imagem e análise de contornos da imagem - **numpy**- com um excelente manipulador de matrizes - **matplotlib** - para inserção de números dentro da imagem.

2 Metodologia e Análise de Resultados

Para atender às requisições propostas pelo enunciado, o código **trab1.py** foi desenvolvido, escrito em linguagem Python. O código é composto de 5 partes em busca de praticidade para desenvolver e explicar cada passo tomado pelo autor na construção de seu código. A primeira e mais simples parte é a leitura e abertura da imagem, na qual a imagem é armazenada como uma matriz com a posições dos pixels e as cores RGB inclusas.

A segunda parte do código corresponde na transformação da imagem para trocar seu rgb para tons de cinzas. Para isso se utilizou as funções `rgb2grey`, da classe `color` encontrada na biblioteca **skimage**, e a função `invert` da classe `util` também na biblioteca **skimage**. Segue-se embaixo o resultado obtido ao usar como entrada as Figuras 1 e 2.

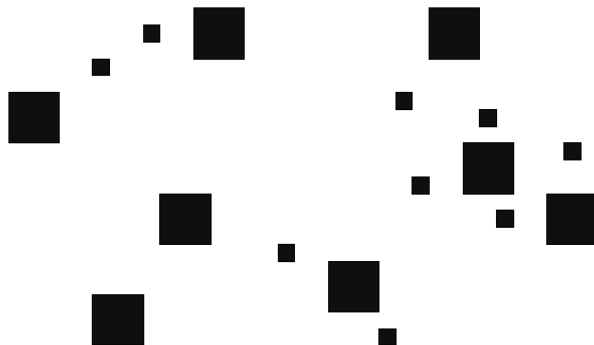


Figure 3: Imagem 'objetos1Cinza.png' obtida como resultado do programa

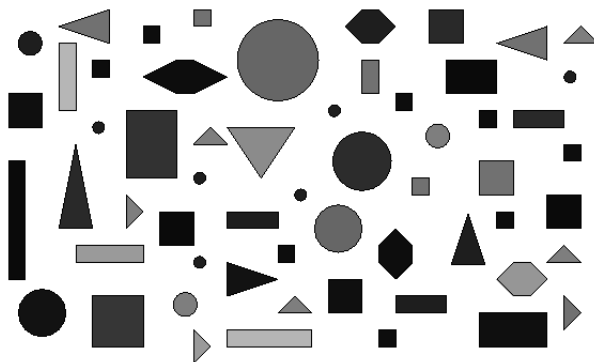


Figure 4: Imagem 'objetos2Cinza.png' obtida como resultado do programa

Em sequência, para resolver o item 1.2 proposto pelo enunciado do projeto, foi escrito no código a seção que imprime o contorno de cada imagem identificada. Para isso, utilizou-se uma importante função da mesma biblioteca **skimage**: a função **find_contours**, capaz de identificar todos os contornos dos objetos observados pela imagem. Segue-se os resultados obtidos com esse método:

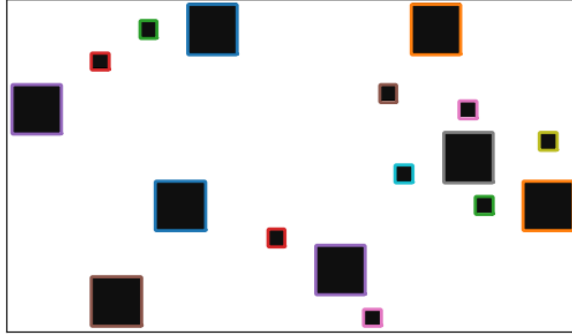


Figure 5: Imagem 'objetos1Contorno.png' obtida como resultado do programa

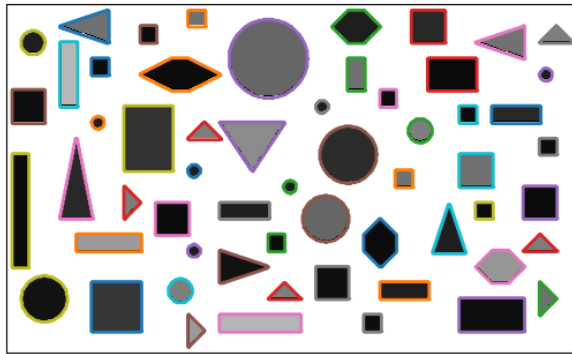


Figure 6: Imagem 'objetos2Contorno.png' obtida como resultado do programa

A quarta divisão do programa é responsável pela resolução proposta pelo aluno para item 1.3. Utilizando das funções `label`, responsável por rotular as imagens e sua separação para melhor identificação e manuseamento em termos de cálculos de suas propriedades, e `regionprops`, responsável pelos cálculos em dada região rotulada, encontradas na classe **measure** na biblioteca **skimage**, foram construídas as seguintes imagens:

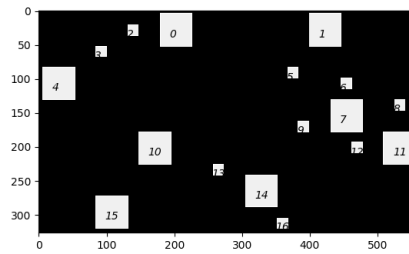


Figure 7: Imagem 'objetos1Rotulado.png' obtida como resultado do programa

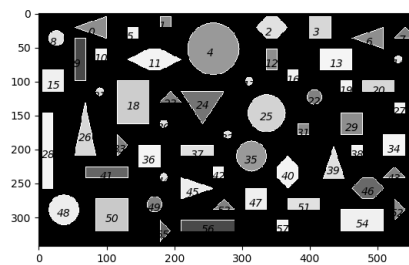


Figure 8: Imagem 'objetos2Rotulado.png' obtida como resultado do programa

Além das imagens, as propriedades numéricas de cada imagem rotulada foram impressas nas seguintes tabelas:

| Item 1.3 - Propriedades do Objeto | | | |
|-----------------------------------|----|------------|----------------|
| 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 |

Figure 9: Imagem 'imagem1Props.png' obtida como resultado do programa

```

Digite o nome da imagem:objetos2
Item 1.3 - Propriedades do Objeto
regiao: 0 perimetro: 138 area: 126
regiao: 1 perimetro: 62 area: 272
regiao: 2 perimetro: 118 area: 94
regiao: 3 perimetro: 126 area: 1056
regiao: 4 perimetro: 251 area: 214
regiao: 5 perimetro: 62 area: 272
regiao: 6 perimetro: 136 area: 125
regiao: 7 perimetro: 77 area: 64
regiao: 8 perimetro: 73 area: 63
regiao: 9 perimetro: 158 area: 1088
regiao: 10 perimetro: 64 area: 289
regiao: 11 perimetro: 181 area: 157
regiao: 12 perimetro: 94 area: 544
regiao: 13 perimetro: 156 area: 1536
regiao: 14 perimetro: 36 area: 31
regiao: 15 perimetro: 126 area: 1056
regiao: 16 perimetro: 62 area: 272
regiao: 17 perimetro: 36 area: 31
regiao: 18 perimetro: 220 area: 3072
regiao: 19 perimetro: 64 area: 289
regiao: 20 perimetro: 126 area: 816
regiao: 21 perimetro: 37 area: 32
regiao: 22 perimetro: 73 area: 64
regiao: 23 perimetro: 76 area: 64
regiao: 24 perimetro: 183 area: 157
regiao: 25 perimetro: 181 area: 154
regiao: 26 perimetro: 200 area: 188
regiao: 27 perimetro: 62 area: 272
regiao: 28 perimetro: 252 area: 1792
regiao: 29 perimetro: 128 area: 1089
regiao: 30 perimetro: 36 area: 31
regiao: 31 perimetro: 64 area: 289
regiao: 32 perimetro: 37 area: 32
regiao: 33 perimetro: 74 area: 62
regiao: 34 perimetro: 126 area: 1056
regiao: 35 perimetro: 147 area: 126
regiao: 36 perimetro: 126 area: 1056
regiao: 37 perimetro: 126 area: 784
regiao: 38 perimetro: 62 area: 272
regiao: 39 perimetro: 138 area: 126
regiao: 40 perimetro: 118 area: 94
regiao: 41 perimetro: 158 area: 1088
regiao: 42 perimetro: 62 area: 272
regiao: 43 perimetro: 77 area: 64
regiao: 44 perimetro: 36 area: 31
regiao: 45 perimetro: 137 area: 125
regiao: 46 perimetro: 118 area: 94
regiao: 47 perimetro: 124 area: 1024
regiao: 48 perimetro: 146 area: 126
regiao: 49 perimetro: 73 area: 63
regiao: 50 perimetro: 192 area: 2401
regiao: 51 perimetro: 126 area: 816
regiao: 52 perimetro: 77 area: 64
regiao: 53 perimetro: 74 area: 62
regiao: 54 perimetro: 190 area: 2112
regiao: 55 perimetro: 74 area: 62
regiao: 56 perimetro: 190 area: 1360
regiao: 57 perimetro: 64 area: 289

```

Figure 10: Imagem 'imagem2Props.png' obtida como resultado do programa

Na última seção do código é construído o histograma responsável por separar as imagens em 3 regiões: pequenas, médias e grandes, de acordo com o critério exposto no enunciado. Para a produção do histograma utilizou-se a biblioteca **matplotlib** e os resultados foram:

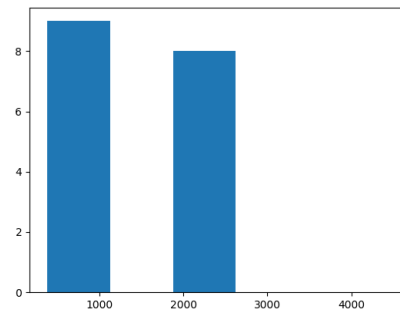


Figure 11: Imagem 'objetos1histograma.png' obtida como resultado do programa

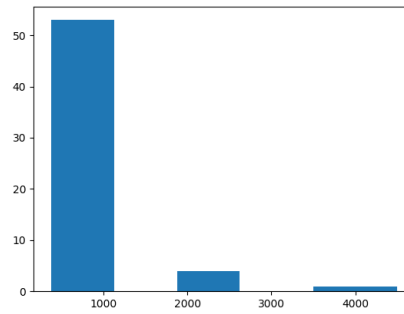


Figure 12: Imagem 'objetos2histograma.png' obtida como resultado do programa

3 Conclusão

Durante a construção do código responsável pela solução dos problemas proposto pelo projeto, o aluno encontrou diversas maneiras de solucionar os problemas com diferentes bibliotecas, implicando nas diferentes soluções que podem ser desenvolvidas. Construções anteriores do programa utilizavam a biblioteca PIL para rotular as regiões, entretanto houveram diferentes conflitos com o uso dessa biblioteca e as matrizes de imagens em numpy.

Além disso, a aplicação de conceitos teóricos em projetos práticos exigem uma análise cuidadosa da documentação das bibliotecas manipuladora de imagens, devido a grande e diversa quantidade de características que podem alterar profundamente o resultado que você pode encontrar. Isso foi observado durante a construção dos histogramas e ao tentar salvar as imagens que eram alteradas utilizando a função subplot da biblioteca matplotlib.