

## 1. Introdução

Neste relatório são apresentados métodos para extração de contornos de objetos, bem como suas respectivas medidas de área, perímetro, excentricidade e solidez em imagens digitais. Para tal propósito, foram utilizadas as bibliotecas *Numpy*, *OpenCV* e *Scikit Image* em Python.

## 2. Imagens utilizadas

Abaixo estão dispostas as duas imagens utilizadas neste experimento:

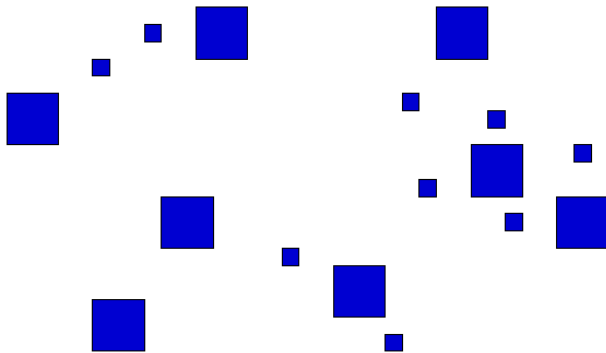


Figura 1: Quadrados



Figura 2: Objetos

## 3. Transformação de Cores e Binarização

A fim de facilitar o processo de extração dos contornos dos objetos de imagens digitais, convertemos o espaço de cores das mesmas para níveis de cinza e realizamos a binarização. A conversão do espaço de cores é realizada com o método *cvtColor* da biblioteca *OpenCV*, enquanto a binarização é realizada pela biblioteca *Numpy*, mais especificamente com o método *where*. A conversão do espaço de cores e binarização é realizada dentro da função *contornos*.

É possível visualizar abaixo o resultado da etapa descrita acima:

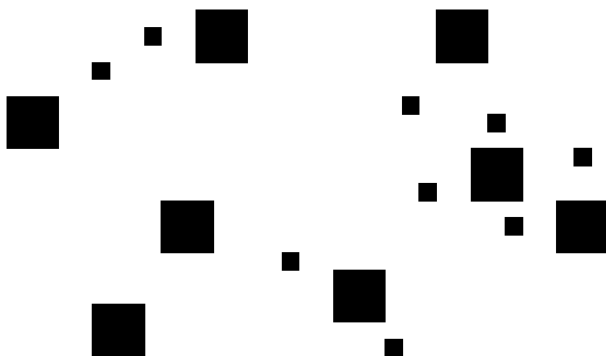


Figura 3: Quadrados, binarizada



Figura 4: Objetos, binarizada

#### 4. Contorno dos Objetos

Após a etapa de conversão do espaço de cores e binarização, estamos aptos para encontrar os contornos dos objetos presentes nas imagens. Esta etapa foi realizada com auxílio do método *findContours*, da biblioteca *OpenCV*. O método *findContours* retorna um *array* contendo outros *arrays*, as coordenadas dos contornos de cada um dos objetos identificados na imagem.

Após a execução do método citado acima, podemos então, obter o desenho dos contornos. Para tal tarefa, utiliza-se o método *drawContours*, também da biblioteca *OpenCV*. O método toma como parâmetro, a imagem na qual será desenhada o resultado dos contornos, dessa forma, passa-se como parâmetro a própria imagem original.

O procedimento citado acima, no entanto, faz com que tenhamos um fundo preto nos seus resultados. A fim de evitar isso, utilizou-se novamente o método *where*, para mudar o fundo preto para branco. Finalmente, podemos observar o resultado dos procedimentos descritos acima:

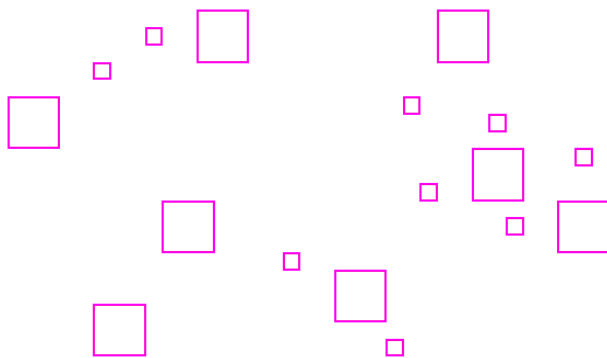


Figura 5: Contorno de Quadrados



Figura 6: Contorno de Objetos

#### 5. Extração de Propriedades dos Objetos

Nesta seção haverá subseções de descrição de cada um dos métodos de extração de propriedades dos objetos. De modo geral, a extração de todas as propriedades se deu dentro de um *loop* pelo *array* retornado pelo método *regionprops* da biblioteca *Scikit Image*. Este *array* contém objetos que representam cada uma das formas presentes nas imagens, contendo vários atributos, inclusive os quais estamos procurando.

Através do método *regionprops*, a obtenção das propriedades desejadas foi realizada com facilidade e eficiência. Aproveitou-se o fato de que o centróide é um dos atributos contidos em todos os objetos do *array* de retorno do método para inserir um número de identificação em cada um dos objetos presentes nas imagens.

Todos os resultados são apresentados abaixo, bem como um histograma de frequência de objetos pequenos, médios ou grandes, sendo que objetos pequenos possuem uma área de

menos de 1500 pixels; um objeto médio possui área entre 1500 e 3000 pixels e; um objeto grande tem área de mais de 3000 pixels.

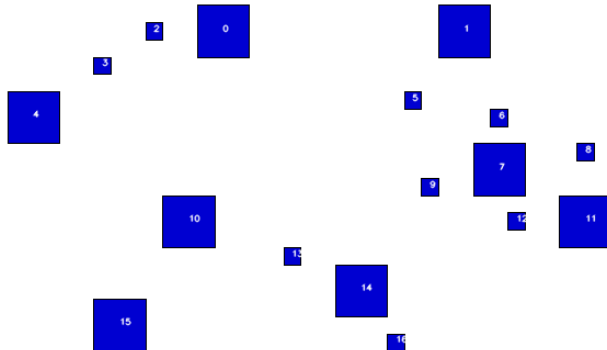


Figura 7: Quadrados numerados



Figura 8: Objetos numerados

#### Quadrados:

região 0:	área: 2352	perímetro: 190.0	solidez: 1.0	excentricidade: 0.201039
região 1:	área: 2352	perímetro: 190.0	solidez: 1.0	excentricidade: 0.201039
região 2:	área: 272	perímetro: 62.0	solidez: 1.0	excentricidade: 0.338502
região 3:	área: 272	perímetro: 62.0	solidez: 1.0	excentricidade: 0.338502
região 4:	área: 2304	perímetro: 188.0	solidez: 1.0	excentricidade: 0.0
região 5:	área: 272	perímetro: 62.0	solidez: 1.0	excentricidade: 0.338502
região 6:	área: 289	perímetro: 64.0	solidez: 1.0	excentricidade: 0.0
região 7:	área: 2352	perímetro: 190.0	solidez: 1.0	excentricidade: 0.201039
região 8:	área: 289	perímetro: 64.0	solidez: 1.0	excentricidade: 0.0
região 9:	área: 289	perímetro: 64.0	solidez: 1.0	excentricidade: 0.0
região 10:	área: 2352	perímetro: 190.0	solidez: 1.0	excentricidade: 0.201039
região 11:	área: 2352	perímetro: 190.0	solidez: 1.0	excentricidade: 0.201039
região 12:	área: 289	perímetro: 64.0	solidez: 1.0	excentricidade: 0.0
região 13:	área: 272	perímetro: 62.0	solidez: 1.0	excentricidade: 0.338502
região 14:	área: 2304	perímetro: 188.0	solidez: 1.0	excentricidade: 0.0
região 15:	área: 2352	perímetro: 190.0	solidez: 1.0	excentricidade: 0.201039
região 16:	área: 272	perímetro: 62.0	solidez: 1.0	excentricidade: 0.338502

número de regiões pequenas: 9

número de regiões médias: 8

número de regiões grandes: 0

#### Objetos:

região 0:	área: 3969	perímetro: 313.764502	solidez: 0.746614	excentricidade: 0.816362
região 1:	área: 791	perímetro: 119.982756	solidez: 0.89581	excentricidade: 0.741103
região 2:	área: 3584	perímetro: 259.462987	solidez: 0.975504	excentricidade: 0.898073
região 3:	área: 540	perímetro: 99.254834	solidez: 0.901503	excentricidade: 0.889586

região 4: área: 438 perímetro: 88.769553 solidez: 0.908714 excentricidade: 0.855923  
 região 5: área: 1684 perímetro: 174.124892 solidez: 0.96726 excentricidade: 0.868169  
 região 6: área: 642 perímetro: 103.012193 solidez: 0.962519 excentricidade: 0.890242  
 região 7: área: 3934 perímetro: 305.421356 solidez: 0.773192 excentricidade: 0.910992  
 região 8: área: 675 perímetro: 96.325902 solidez: 0.969828 excentricidade: 0.62038

número de regiões pequenas: 5  
 número de regiões médias: 1  
 número de regiões grandes: 3

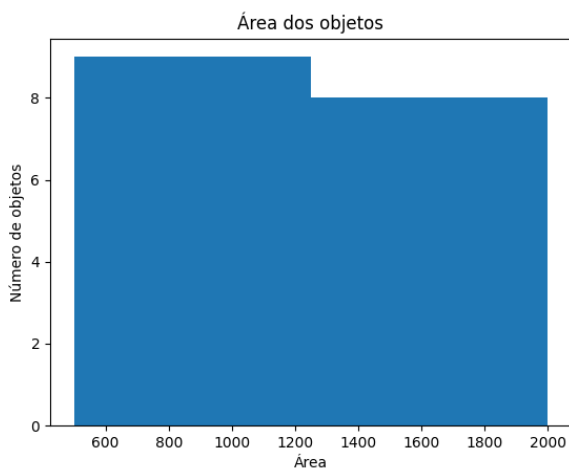


Figura 9: Histograma de área dos objetos de Quadrados

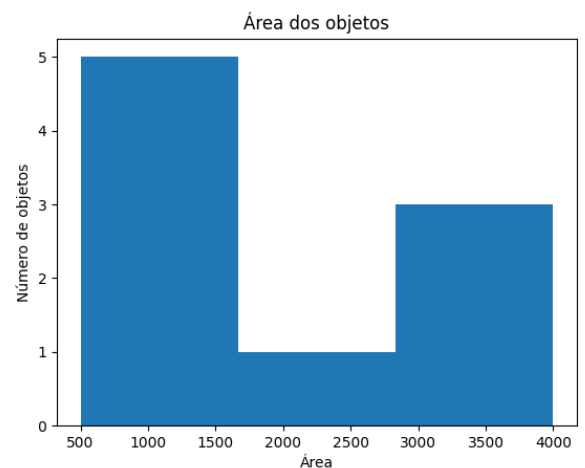


Figura 10: Histograma de área de Objetos

Dessa forma, concluímos que o processo de obtenção dos contornos, bem como a extração das propriedades de objetos presentes nas imagens digitais, mostrou-se bem sucedido, apresentando resultados satisfatórios e condizentes com o esperado (objetos visivelmente menores apresentando menores áreas, por exemplo). A codificação do processo foi bem direta e eficiente, graças às ferramentas presentes em bibliotecas como *OpenCV* e *Scikit Image*.