

# Relatorio PIM

Amyr Allan

## Abstract

Solução do trabalho de segmentação por área. Essa envolve a implementação de uma pequena biblioteca utilizada para estudar a morfologia de imagens. A descrição das funções implementadas pode ser verificada no repositório do trabalho.

## 1 Descrição

### 1.1 Equipe

A equipe desse presente trabalho é composta pelo aluno Amyr Allan.

### 1.2 Repositório

Todo o código do trabalho pode ser verificado em [github.com/Amyr14/segmentation](https://github.com/Amyr14/segmentation)

### 1.3 Tarefa

A solução apresentada nas sessões abaixo é para a tarefa de segmentação por área. A entrada é uma imagem em tons de cinza composta por diversos objetos com diferentes brilhos e tamanhos. Nessa, existe um objeto de interesse que se apresenta como uma grande "ilha de pixels". Deve-se limiarizar e segmentar a imagem de forma a evidenciar esse objeto, finalmente calculando as coordenadas do seu "centro de massa".

### 1.4 Dependências

Esse projeto possui como dependências as bibliotecas scikit-image e numpy. Para instala-las, no terminal, basta digitar:

```
pip install scikit-image
pip install numpy
```

### 1.5 Instruções de Execução

Os arquivos de entrada devem ser postos na pasta "inputs". Para executar o programa pelo terminal, digite:

```
python3 main.py arquivo1.png arquivo2.png ...
```

O programa irá executar sob os arquivos passados como argumento. A saída (coordenadas do centróide) será apresentada no terminal.

## 2 Solução

A solução proposta envolve o uso de algoritmos de limiarização, identificação de componentes conexos e identificação de objetos pequenos e de borda.

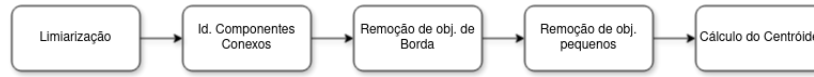


Figure 1: Pipeline da solução

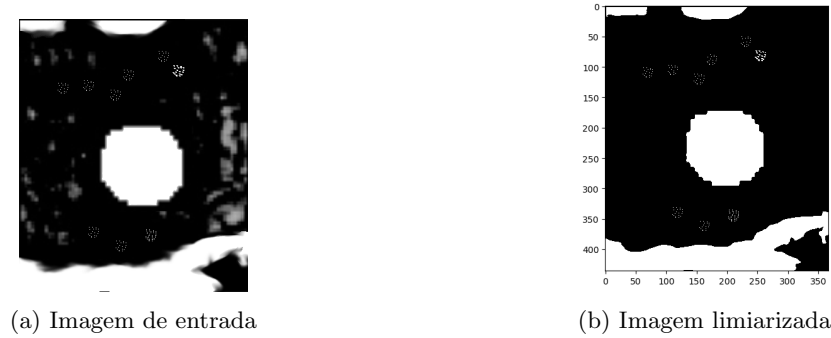


Figure 2: Antes e depois do processo de limiarização; Note a grande redução de ruído

## 2.1 Limiarização

Primeiramente, utilizou-se o algoritmo multiotsu com quatro classes para o cálculo do limiar, resultando no retorno de três limiares. Por meio de testes heurísticos, o limiar mais alto foi identificado como o mais adequado para limiarizar a imagem de entrada.

## 2.2 Identificação de Componentes Conexos

Em seguida, uma função de implementação própria chamada `get_connected_components` foi utilizada para identificar os componentes conexos na imagem binária. Dessa forma, obteve-se uma matriz (mesmas dim. da imagem) e uma lista de labels.

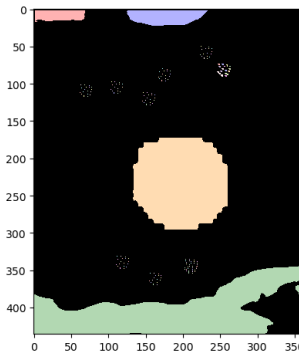


Figure 3: Imagem segmentada

## 2.3 Remoção de Componentes Conectados a Borda

Em uma primeira análise, percebe-se que existe um grande objeto conectado com a borda. Tendo isso em vista, utilizou-se uma função de implementação própria chamada `get_border_components` para identifica-lo. Essa função recebe a matriz de labels e retorna as labels dos objetos de borda. A partir dessa informação, os objetos foram removidos em seguida por meio de uma máscara booleana.

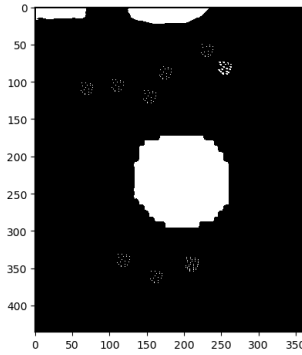


Figure 4: Imagem sem componentes de borda

## 2.4 Remoção de Componentes Pequenos

Ao analisar-se os objetos restantes, percebe-se que o objeto de interesse é consideravelmente maior que os outros. Dessa forma, pode-se filtrar todos os objetos que possuam menos que uma certa quantidade de pixels. Assim, utilizou-se uma função de implementação própria chamada `get_small_components`. Essa função recebe uma matriz de labels e uma quantidade mínima de pixels, retornando as labels dos objetos que possuem menos pixels do que a quantidade mínima. Como no último passo, utilizou-se dessa informação para a remoção desses objetos por meio de uma máscara booleana.

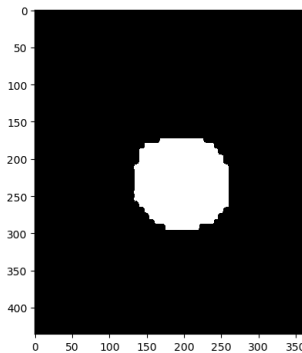


Figure 5: Imagem sem componentes pequenos

## 2.5 Cálculo do Centróide

Olhando para a imagem resultante, nota-se que somente o objeto de interesse permanece. Logo, pode-se simplesmente recuperar as coordenadas dos pixels ligados e calcular a média para os eixos X e Y. Segue um exemplo de código abaixo.

```
indices = numpy.nonzero(imagem_binaria)
centroide = numpy.around(numpy.mean(indices, axis=1))
```

## 3 Resultado

Por fim, após todas essas etapas, obteve-se o centróide  $\{233, 197\}$ . A partir de uma inspeção visual, pode-se conferir que essa é uma boa aproximação do centro de massa do objeto de interesse.