

Relatório Projeto 02

Introdução ao Processamento de Imagens

1st Gabriel Martins de Almeida

Matricula: 190013371

Brasília, brasil

gbmalme@gmail.com

I. INTRODUÇÃO

A morfologia matemática é uma técnica de processamento de imagens que se baseia em operações matemáticas para analisar e manipular a forma e a estrutura de objetos em uma imagem. A morfologia matemática utiliza operações fundamentais, como dilatação, erosão, abertura e fechamento, que são aplicadas a uma imagem por meio de um elemento estruturante. Essas operações são utilizadas para extrair informações relevantes sobre a forma, tamanho, conectividade e distribuição espacial dos objetos presentes na imagem. O projeto 2 da disciplina introdução ao processamento de imagens visa implementar a solução para três problemas com essas técnicas.

II. METODOLOGIA

Este Relatório apresenta os métodos utilizados e resultados obtidos na implementação do projeto. No desenvolvimento foi utilizado a linguagem Python na versão 3.9 com as bibliotecas cv2, numpy e scipy para realizar as operações com imagens.

III. RESULTADOS

A. Questão 1

A questão apresenta uma imagem chamada pcb.jpg que representa um sistema de inspeção de circuitos, o problema consiste em implementar um programa que verifique o diâmetro em pixels e a quantidade de buracos de um circuito baseado na imagem.



Fig. 1. Imagem pcb.jpg Original

Para solucionar o problema foi inicialmente feito uma binarização na imagem e em seguida uma operação de fechamento, utilizando o elemento estruturante elipse. O resultado é a imagem apresentada abaixo.



Fig. 2. Imagem após operação de fechamento

Em seguida é feita a operação de preenchimento na imagem, foi utilizada a função “binary fill holes” da biblioteca scipy, o que resulta na remoção dos círculos desconectados, conforme apresentado na imagem abaixo.



Fig. 3. Imagem após operação de preenchimento

O próximo passo é utilizar a imagem gerada pela operação de preenchimento como uma máscara que será aplicada na imagem obtida pelo fechamento e para realizar essa operação é necessário realizar a diferença entre a imagem preenchida e a imagem após o fechamento. O resultado será a imagem apresentada abaixo.

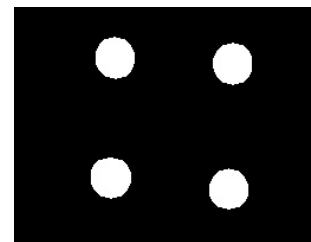


Fig. 4. Diferença de imagens entre preenchimento e o fechamento

Com a imagem obtida anteriormente em mãos é possível calcular a quantidade de buracos e seus respectivos diâmetros.

Para fazer isso foi necessário utilizar a função “findContours” do OpenCV, que retorna uma lista com os contornos encontrados na imagem e a partir disso é possível obter a quantidade de buracos. Com a lista de buracos em mãos é possível utilizar a função “contourArea” do openCV para calcular a área dos buracos e com a fórmula apresentada abaixo é possível calcular o diâmetro dos buracos.

$$Diametro = \sqrt{\frac{4 \cdot Area}{\pi}}$$

O resultado está expresso na imagem abaixo.

```
Quantidade de buracos: 4
Diametro dos buracos:
32.9556267071208
33.76664052449308
33.549124542167775
33.49214891450948
```

Fig. 5. Resultado da questão 01

B. Questão 2

A questão apresenta uma imagem chamada morf.teste.png e o objetivo é transformar a mesma em uma imagem binária, onde o fundo é branco e as imagens são pretas. Logo, é necessário isolar as letras do fundo.

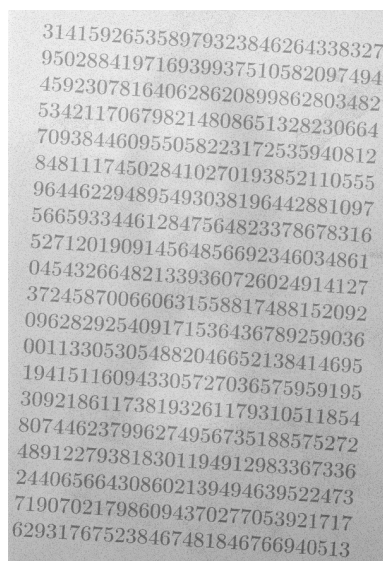


Fig. 6. Imagem morf.teste.png Original

O primeiro passo para solucionar o problema foi realizar a operação de fechamento utilizando o elemento estruturante elipse, para isolar o fundo da imagem e o resultado obtido foi a imagem abaixo.



Fig. 7. Imagem após o fechamento.

Então com o resultado do fechamento foi possível realizar a diferença entre a imagem original e o fundo obtido, o resultado é a imagem abaixo.



Fig. 8. Imagem obtida da diferença

Com o intuito de melhorar o resultado obtido anteriormente, foi aplicado um filtro black hat na imagem original e segue o resultado abaixo.



Fig. 9. Imagem original com filtro black hat

No resultado obtido com o filtro black hat, com o elemento estruturante elipse (1,1), foi realizada as operações de binarização, dilatação, erosão e fechamento nessa ordem com o intuito de remover símbolos desconectados ou ruídos, por fim foi realizada a inversão de cores na imagem e o resultado é a imagem a seguir.

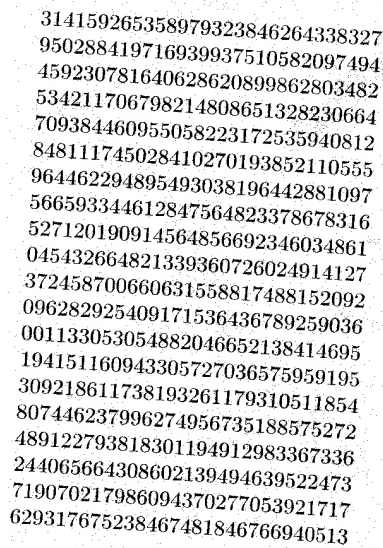


Fig. 10. Resultado questão 02

C. Questão 3

Está questão da a imagem img.cells.jpg e é necessário seguir uma série de passos cujo objetivo é detectar a borda da imagem recebida.

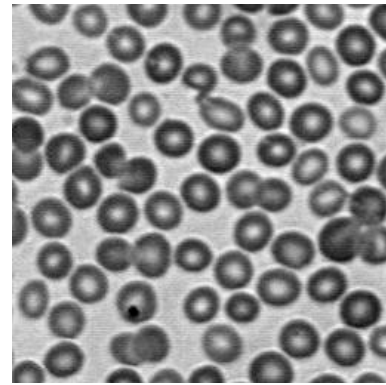


Fig. 11. Imagem Original img.cells.jpg

Sendo assim o primeiro passo é realizar a binarização da imagem, visando uma imagem com as células pretas e o fundo branco. Para fazer isso foi feita uma binarização e em seguida uma operação de fechamento para otimizar o resultado, segue abaixo a imagem obtida.

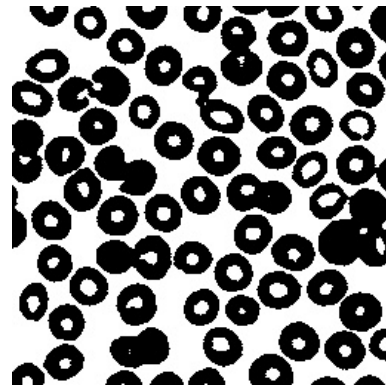


Fig. 12. Imagem Binarizada

Entretanto, a imagem resultante da binarização ainda possui uma má segmentação e para corrigir isso foi realizada uma operação de preenchimento, o resultado está a seguir.

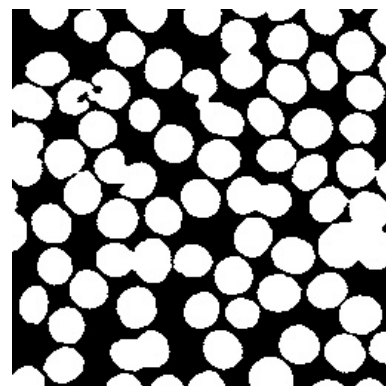


Fig. 13. Imagem após preenchimento

Com a imagem obtida do preenchimento é realizada uma dilatação com o elemento estruturante matriz (3,3), em seguida

é calculada a função de distância da imagem e o resultado é a imagem abaixo.

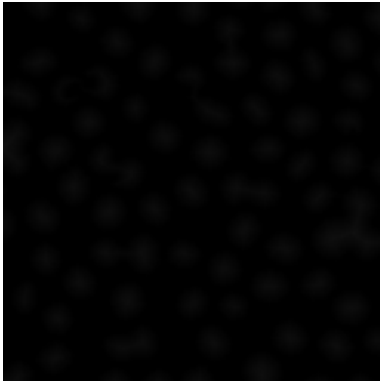


Fig. 14. Imagem função de distância

Por fim é computada a segmentação “Watershed” da imagem e o resultado dessa operação está descrito na imagem abaixo.

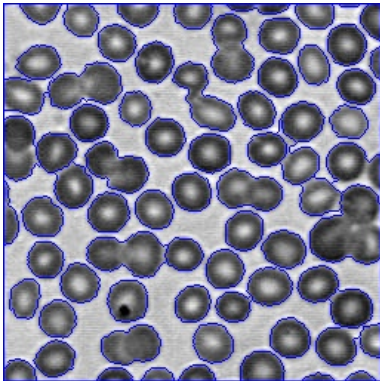


Fig. 15. Resultado da questão 03

O resultado foi satisfatório se tratando de bordas, mas falhou na identificação dos objetos, pois há células se sobrepondo e sendo segmentadas de forma errada. a imagem função de distância ficou muito escura no relatório, mas no programa foi possível visualizar as marcas de cada célula, o que ajuda a entender a lógica dos algoritmos.

IV. CONCLUSÕES

Após a conclusão do projeto foi possível observar como utilizar os métodos de morfologia matemática na prática.