Danillo Rodrigues Abreu

Projeto de Processamento de Imagens Operações Morfológicas *Dilatação, Erosão, Abertura e Fechamento*

Arapiraca – AL
2021

Danillo Rodrigues Abreu

Projeto de Processamento de Imagens
Operações Morfológicas *Dilatação, Erosão, Abertura e Fechamento*

Projeto solicitado aos discentes do 7º período do curso Ciência da Computação, para fins avaliativos da disciplina de Processamento de Imagens.

Universidade Federal de Alagoas – UFAL
Campus Arapiraca
Ciência da Computação

Orientador: Prof. Dr. Tácito Trindade de Araújo Tiburtino Neves

Arapiraca – AL
2021

Sumário

Projeto de Processamento de Imagens

1 Descrição do projeto

O presente projeto foi desenvolvido com a tecnologia: Linguagem de programação Python 3.9.5 e as bibliotecas PIL e numpy. Ele tem o intuito de implementar um código que permita interação do usuário na escolha de um elemento estruturante e que receba uma imagem de entrada. A saída do algoritmo será o resultado da aplicação do elemento estruturante para as operações de morfologia: dilatação, erosão, abertura e fechamento.

2 Implementação das funcionalidades diversas do projeto

2.1 Conversão da imagem em escala cinza

Uma imagem é uma matriz de pixels onde cada elemento representa a intensidade luminosa média, I(x,y), correspondente aos pontos da cena adquirida. Dessa forma é possível alterar cada pixel dessa matriz afim de converter uma determinada imagem.

Ao converter uma imagem RBG colorida em uma imagem de tons de cinza temos a ”simplificação” dessa imagem, visto que, uma imagem RGB normalizada tem: 3 X 8 bits por plano de cor, I R (x,y), I G (x,y), I B(x,y). Enquanto que uma imagem em tons de cinza tem: 8 bits por pixel no intervalo de [0 a 255].

O próximo passo, na morfologia, depois da convenção de imagens em tons de cinza é a converção binária (limiarização).

Para realizar a conversão em escala de cinza foi definida a seguinte função:

- *def convertendoImagemEmEscalaCinza(imagem)*: recebe como parâmetro uma imagem, realiza a conversão e retorna a imagem em tons de cinza

Resultado obtido ao utilizar a função de converter imagem em escala de cinza:

Figura 1 – Imagem original.

Fonte: https://bit.ly/3vb6501

Figura 2 – Imagem em escala de cinza.

Fonte: dados do projeto.

2.2 Limiarização ou binarização

Na morfologia, antes de aplicar as operações de erosão e dilatação em imagens, far-se-à necessário aplicar a limiarização nessas.

A limiarização consiste em separar uma imagem, em regiões de interesse e não interesse através da escolha de um ponto de corte. Essas regiões podem ser representadas por pixels pretos e brancos. Os métodos mais simples de limiarização utilizam um único ponto de corte também conhecido por WKUHVKROG.

No projeto é feito a conversão da imagem, em níveis de cinza, para uma imagem com representação binária (dois tons). Esse processo de conversão é importante para uma série de objetivos, tais como:

- Identificar objetos e separá-los do fundo da imagem;
- Quando analisar a forma da imagem é mais importante que a intensidade dos píxeis;
- Apresentar a imagem em um dispositivo de saída que tem somente um bit de resolução de intensidade, ou seja, um dispositivo de dois níveis, como uma impressora.

Para realizar a limiarização de imagens foi definida a seguinte função:

- *def aplicarLimiarizacaoDeImagem(imagem)*: recebe como parâmetro uma imagem e aciona a função: *def convertendoImagemEmEscalaCinza(imagem)* para converter a imagem recebida em uma uma imagem em níveis de cinza. Já tendo em ”mãos” a imagem em níveis de cinza é feito a analise do seu histograma para descreve a distribuição estatística dos níveis de cinza em termos do número de amostras (pixels) com cada nível. Após a limiarização, a imagem resultante terá apenas dois tons (preto e branco).

Resultado obtido ao utilizar a função de limiarização:

Figura 3 – Imagem em escala de cinza

Fonte: dados do projeto

Figura 4 – Imagem limiarizada

Fonte: dados do projeto

3 Implementação dos requisitos solicitados

3.1 Considerações sobre erosão e dilatação

O complemento de uma erosão é o mesmo que uma dilatação do complemento da imagem pelo elemento estruturante refletido, tal que:

- A dilatação expande uma imagem e a erosão reduz;
- Erosão e dilatação não são transformações inversas, se a uma imagem se faz uma erosão seguida de uma dilatação o resultado não é a imagem incial, em vez disso o resultado é uma versão simplificada e menos detalhada do que a original;
- Erosão e dilatação são operações duais:

$$(A \Theta B)^c = A^c \oplus \hat{B}$$

3.2 Erosão

A primeira operação básica da morfologia matemática é a erosão de imagens, ela é baseada na erosão do solo, na qual, o solo é corroído e desgastado por dirvesos fatores naturais e/ou provocados. Na erosão de imagens são corroídos os limites do objeto em primeiro plano. Como as imagens estão em dois tons (preto e branco), graças ao processo de limiarização, a erosão afeta o tom branco da imagem, fazendo o desgaste do mesmo. Efeitos da erosão:

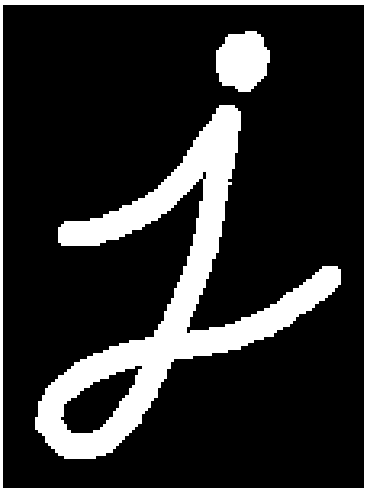
- Diminuir partículas;
- Eliminar componentes menores que o elemento estruturante;
- Aumentar buracos;
- Permitir a separação de componentes conectados.

Para aplicar a funcionalide, dentro do projeto, referente a operação morfológica de erosão foi definido uma função:

- *def aplicarErosaoNaImagem(imagem)*: recebe como parâmetro uma imagem já limiarizada pela função: *def aplicarLimiarizacaoNaImagem(imagem)*, então aplica-se a operação de erosão nessa imagem e a retonar.

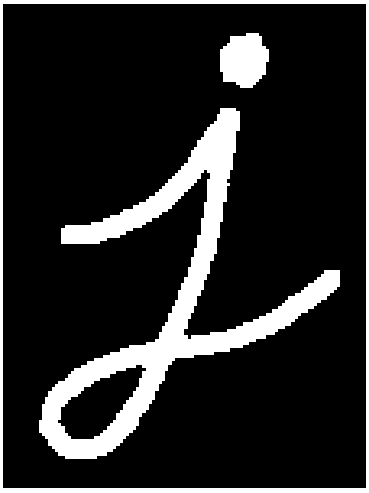
Resultado obtido ao utilizar a função de erosão:

Figura 5 – Imagem limiarizada



Fonte: dados do projeto

Figura 6 – Imagem que sofreu erosão



Fonte: dados do projeto

3.3 Dilatação

Na dilatação um elemento pixel é ”1” se pelo menos um pixel sob o kernel for ”1”. Assim, aumenta a região branca na imagem ou tamanho do objeto em primeiro plano aumenta. Normalmente, em casos como remoção de ruído, a erosão é seguida de dilatação. Porque a erosão remove ruídos brancos, mas também encolhe nosso objeto. Então dilatarmos. Como o barulho se foi, eles não voltarão, mas nossa área de objeto aumenta. Também é útil para unir partes quebradas de um objeto. Efeitos da Dilatação:

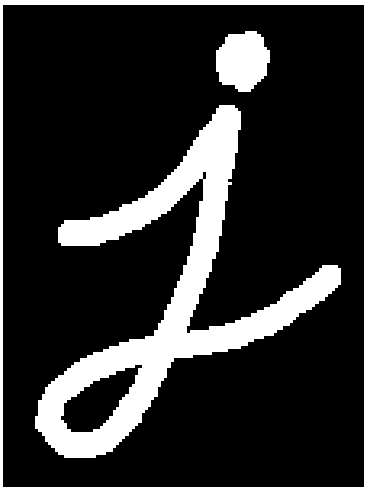
- Aumenta as partículas;
- Preenche buracos negros;
- Conecta grãos próximos;

Para aplicar a funcionalide, dentro do projeto, referente a operação morfológica de dilatação foi definido uma função:

- *def aplicarDilatacaoNaImagem(imagem)*: recebe como parâmetro uma imagem já limiarizada pela função: *def aplicarLimiarizacaoNaImagem(imagem)*, então aplica-se o operação de dilatação nessa imagem e a retonar.

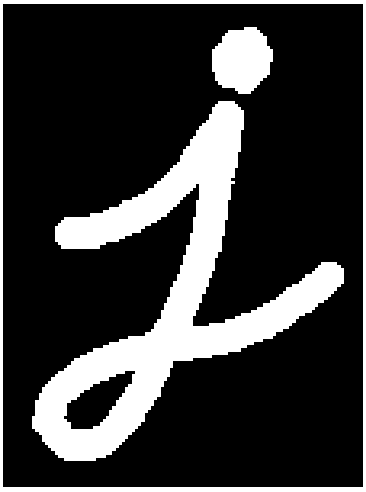
Resultado obtido ao utilizar a função de dilatação:

Figura 7 – Imagem limiarizada



Fonte: dados do projeto

Figura 8 – Imagem que sofreu dilatação



Fonte: dados do projeto

3.4 Abertura

Abertura é a operação morfológica que consiste em aplicar a operação erosão seguida da operação dilatação. É útil na remoção do ruído branco como já explicado acima.

Para aplicar a funcionalide, dentro do projeto, referente a operação morfológica de abertura foi definido uma função:

- *def abertura(imagem)*: recebe como parâmetro uma imagem já limiarizada pela função: *def aplicarLimiarizacaoNaImagem(imagem)*, então aplica-se a operação de erosão nessa imagem, e posteiormente, já na imagem resultante da erosão, aplica-se a operação de dilatação nessa imagem e a retonar.

Resultado obtido ao utilizar a função de abertura:

Figura 9 – Imagem limiarizada

Fonte: dados do projeto

Figura 10 – Imagem que sofreu abertura

Fonte: dados do projeto

3.5 Fechamento

O fechamento é a operação morfológica que consiste em aplicar a operação dilatação seguida da operação erosão. Ele serve para remover ruídos dentro de objetos em primeiro plano, ou pequenos pontos pretos no objeto.

Para aplicar a funcionalide, dentro do projeto, referente a operação morfológica de fechamento foi definido uma função:

- *def fechamento(imagem)*: recebe como parâmetro uma imagem já limiarizada pela função: *def aplicarLimiarizacaoNaImagem(imagem)*, então aplica-se a operação de dilatação nessa imagem, e posteiormente, já na imagem resultante da dilatação, aplica-se a operação de erosão nessa imagem e a retonar.

Resultado obtido ao utilizar a função de fechamento:

Figura 11 – Imagem limiarizada

Fonte: dados do projeto

Figura 12 – Imagem que sofreu fechamento

Fonte: dados do projeto

4 Interação

A interação do usuário com o projeto acontece da seguinte forma:

1) O usuário, ao executar o projeto, digita o nome da imagem que ele deseja manipular (é importante lembrar que a imagem escolhida deve se encontrar dentro da pasta ”imagens” do projeto);

2) Feito isso, o usuário agora deve digitar um número relacionado a qual operação ele deseja realizar com a imagem:

   - **1 - Aplicar Limiarização na imagem;**
   - **2 - Aplicar Dilatação na imagem;**
   - **3 - Aplicar Erosão na imagem;**

- **4 - Fazer Abertura da imagem;**
- **5 - Fazer Fechamento da imagem;**

Ao realizar esses dois passos vai ser apresentado na tela uma sequencia de três imagens, na qual cada uma delas é uma parte do processo até o resultado final escolhido nas opções.

5 Executando o projeto

O código fonte do projeto se encontra no GitHub por meio do endereço: <https://github.com/danillobr/PI_Projeto_Operacoes_Morfologicas.git>. Para executá-lo são necessários os seguintes requisitos:

- Python 3.9.5 (ou superior)
- A biblioteca PIL
- A biblioteca numpy

Para instalar as dependências necessárias e executar o projeto siga os passos a seguir:

1) Passo: instale a biblioteca PIL:

   pip install Pillow

2) Passo: instale a biblioteca numpy:

   pip install numpy

3) Passo: execute o comando dentro da pasta principal do projeto:

   python projeto.py

Referências

CHáVEZ, G. C. *Morfologia Matemática*. Universidade Federal de Ouro Preto. Disponível em: <https://bit.ly/2Tbss7t>. Acesso em: 19 maio. 2021.

NETO, R. A. V. *Introdução à Morfologia Matemática Binária e em Tons de Cinza*. Londrina, 2010. Disponível em: <http://www.ime.unicamp.br/~valle/PDFfiles/valente10.pdf>. Acesso em: 20 maio. 2021.

OLIVEIRA, L. E. S. *Processamento de Imagens - Morfologia Matemática Binária*. Universidade Federal do Paraná. Disponível em: <https://www.inf.ufpr.br/lesoliveira/download/morfologia.pdf>. Acesso em: 18 maio. 2021.

VISION, O. S. C. *Morphological Transformations*. docs.opencv.org. Disponível em: <https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html>. Acesso em: 18 maio. 2021.