



UNIVERSIDADE FEDERAL DE
PERNAMBUCO CENTRO DE
INFORMÁTICA
RESIDÊNCIA EM ROBÓTICA E INTELIGÊNCIA ARTIFICIAL

RICARDO CÉSAR MONTEZUMA

2023

1. Diferencie os conceitos de amostragem e quantização no processo de digitalização de uma imagem e a consequência desses processos na imagem digital.

O processo de digitalização de imagens contém duas etapas essenciais, a amostragem e a quantização, na representação de uma imagem contínua em um formato digital. Amostragem, refere-se à captura ou discretização dos valores de uma imagem em intervalos regulares. Nesse processo, um sinal contínuo é discretizado, pegando determinados valores que em seguida irá representar a imagem em um sinal codificado. A imagem é dividida em uma grade de pixels, onde cada pixel representa um ponto discreto na imagem. A amostragem determina a resolução espacial da imagem digitalizada, ou seja, a quantidade de detalhes que podem ser representados. Uma taxa de amostragem mais alta captura mais pontos da imagem, resultando em uma representação mais detalhada. Por outro lado, uma taxa de amostragem mais baixa captura menos pontos, o que pode resultar na perda de detalhes finos.

A quantização refere-se à atribuição de valores discretos a cada ponto de amostragem da imagem digitalizada. Esses valores representam os níveis de intensidade ou a resolução de cor da imagem.

A quantização é realizada definindo-se um número fixo de níveis possíveis para cada componente da imagem, como 8 bits, resultando em 256 níveis. Esses níveis variam de 0 (preto) a 255 (branco). Em uma imagem colorida, a quantização é feita separadamente para cada componente de cor, no RGB, o vermelho, verde e azul. Portanto, enquanto a amostragem lida com a discretização dos pontos da imagem no espaço, criando a matriz referente à imagem e definindo as dimensões da matriz, a quantização define a resolução de cor da imagem, ambos os processos são necessários para converter uma imagem contínua em uma representação digital.

2. No modelo de cor RGB, um tom de cinza é caracterizado por ter valores iguais para R, G e B (por exemplo, $R = G = B = 120$ corresponde a um tom de cinza). O que caracteriza um tom de cinza no modelo de cor CMYK? Prove.

O modelo de cor CMYK, usa as combinações das cores ciano (C), magenta (M) e amarelo (Y), juntamente com o componente preto (K), para representar o modelo e melhorar a reprodução de cores escuras. Para fornecer um maior controle sobre a saturação das cores é utilizado o fator (K).

Para obter um tom de cinza no modelo de cor CMYK, precisamos zerar os valores de C, M, Y e variar a intensidade da cor preta (K). Podemos provar isso, utilizando o $R=G=B=120 \rightarrow 120/255 = 0.47$ e fazendo os seguintes cálculos abaixo:

$$C = 1 - R \quad C = 1 - 0.47$$

$$M = 1 - G \rightarrow M = 1 - 0.47$$

$$Y = 1 - B \quad Y = 1 - 0.47$$

$$K = \min(C, M, Y) = 0.53$$

$$C' = C - K = 0.53 - 0.53 = 0$$

$$M' = M - K = 0.53 - 0.53 = 0$$

$$Y' = Y - K = 0.53 - 0.53 = 0$$

Logo, podemos ver que quando temos qualquer valor de tom de cinza em RGB, o que requer que os valores de R, G e B sejam iguais, ao converter para CMYK, os valores C, M e Y passam a ser 0 e apenas o valor K varia, o que representa a intensidade do preto, e sua variação resulta nos diferentes tons de cinza nesse modelo.

3. Que tipo de informação o histograma de uma imagem pode trazer quanto a uma previsão de possível resultado satisfatório ou não da aplicação de um algoritmo de binarização?

O algoritmo de binarização irá separar os pixels entre preto e branco, de acordo com um limiar. Todos os pixels abaixo do limiar passarão a ser pretos, pois estarão mais próximos do 0, que representa o preto. Já os que estão acima do limiar, passarão a ser brancos, por estarem mais próximos ao valor de branco, representado pelo valor 1. O histograma entrega uma descrição global da aparência da imagem, sendo possível identificar a distribuição de cores, nela contidas. Imagens com alto contraste são facilmente identificadas olhando a distribuição do histograma, quando por exemplo, temos uma concentração de pixels em torno de um determinado valor no eixo x com um pico em y, seguida de uma sequência mais uniforme, seguida de outra concentração mais afastada da primeira no eixo x. Uma distribuição bem balanceada, com uma ampla variedade de intensidades representadas, indica que a imagem possui uma boa diversidade de tons e contrastes. Imagens com contrastes mais bem definidos, são mais fáceis de definir um ponto de corte que retorne uma imagem de maior qualidade após a binarização, evitando que ela fique muito clara ou muito escura, assim preservando melhor a forma da imagem, sem perder muitos detalhes. Isso resulta em uma boa probabilidade de que a aplicação de um algoritmo de binarização seja bem-sucedida, pois a imagem contém informações suficientes para distinguir entre os objetos de interesse e o fundo.

4. Explique os aspectos computacionais a serem considerados na aplicação de uma convolução de um filtro em uma imagem no domínio espacial.

A aplicação de uma convolução de um filtro em uma imagem no domínio espacial envolve vários aspectos computacionais a serem considerados. Aqui estão alguns dos principais aspectos:

1 - Cor não realizável

A convolução, nas suas operações para cada pixel da matriz, há multiplicações e somas que podem passar de 255 ou ficar negativo, o que não é possível no sistema RGB. Precisando ser tratado na hora de implementar um código que faça alguma convolução de um filtro com a matriz de uma imagem. Para evitar que isso ocorra, podemos recortar para a cor mais próxima ou mudar de coordenada no espaço de cor.

2 - Eficiência computacional

um custo computacional quadrático, apenas a percorrer a imagem já tem um $O(N^2)$ afetando a eficiência e o tempo de execução do programa. Assim, o ideal seria tentar percorrer a imagem a menor quantidade de vezes possível.

3 - Extensão do domínio da imagem

A operação de convolução adiciona linhas e colunas na matriz resultante. Essa quantidade de linhas e colunas adicionadas dependem das dimensões da matriz do filtro e tornam a imagem cada vez maior, o que afeta o seu processamento e carregamento, e poderia adicionar informações a mais desnecessárias. Pois ao estender a matriz da imagem em algumas posições, é como se tivéssemos uma borda zero ao redor dela, mas zero é preto, e por isso precisamos levar em conta como iremos estender a imagem. Algumas formas de estender o domínio dessa imagem seriam: Extensão Constante, Nula ou não, Extensão Periódica, Extensão por Reflexão, Não Extensão.

5. Suponha que um filtro Box 3x3 (matriz no slide 111 da aula de filtragem) é usado para processar uma imagem. Em seguida, um filtro laplaciano 3x3 (slide 135 da aula de filtragem) é usado na imagem processada pelo filtro Box. Ou seja, temos:

$$I_{\text{Final}} = \text{Laplaciano}(\text{Box}(I_{\text{Original}}))$$

Calcule como deve ser uma máscara única que faria o mesmo que os filtros Laplaciano e Box. Apresente todos os cálculos.

Para calcular uma máscara que substitua os filtros Laplaciano e Box acima, iremos fazer uma convolução entre os dois filtros e trocá-los pela matriz resultante de dimensões 5x5 que será a máscara única em questão.

A matriz do filtro Box é dada por:

```
1 1 1
1 1 1
1 1 1
```

A máscara do filtro Laplaciano 3x3 é:

0 1 0

1 -4 1

0 1 0

Abaixo, segue os cálculos da convolução entre essas duas máscaras. Como há 25 posições nessa máscara resultante, será mostrado apenas o cálculo das 15 primeiras posições, da seguinte maneira:

- P(1,1) Entrada do filtro na primeira linha e primeira coluna, e início da operação de convolução
- até a P(3, 5) saída da máscara na última posição da terceira linha, a partir de onde começa a quarta linha, seguida da quinta linha. Essas duas linhas possuem respectivamente os mesmo valores da primeira e segunda linha

$P(1,1) = 1 \times 0 = 0$, $P(1,2) = 1 \times 1 + 1 \times 0 = 1$, $P(1,3) = 1 \times 0 + 1 \times 1 + 1 \times 0 = 1$, $P(1,4) = 1 \times 0 + 1 \times 1 = 1$, $P(1,5) = 1 \times 0 = 0$, $P(2,1) = 0 \times 1 + 1 \times 1 = 1$, $P(2,2) = -4 \times 1 + 1 \times 1 + 1 \times 1 + 0 \times 1 = -2$, $P(2,3) = 1 \times 1 + -4 \times 1 + 1 \times 1 + 0 \times 1 + 1 \times 1 + 0 \times 1 = -1$, $P(2,4) = 1 \times 1 + -4 \times 1 + 0 \times 1 + 1 \times 1 = -2$, $P(2,5) = 1 \times 1 + 0 \times 1 = 1$, $P(3,1) = 0 \times 1 + 1 \times 1 + 0 \times 1 = 1$, $P(3,2) = 1 \times 1 + 0 \times 1 + -4 \times 1 + 1 \times 1 + 1 \times 1 + 0 \times 1 = -1$, $P(3,3) = 0 \times 1 + 1 \times 1 + 0 \times 1 + 1 \times 1 + -4 \times 1 + 1 \times 1 + 0 \times 1 + 1 \times 1 + 0 \times 1 = 0$, $P(3,4) = 0 \times 1 + 1 \times 1 + 1 \times 1 + -4 \times 1 + 0 \times 1 + 1 \times 1 = -1$, $P(3,5) = 0 \times 1 + 1 \times 1 + 0 \times 1 = 1$

0 1 1 1 0

1 -2 -1 -2 1

1 -1 0 -1 1

Os cálculos acima são referentes as primeiras 15 posições da máscara resultante, para completar as outras posições, seguiremos a mesma lógica e os valores, nesse caso, serão os mesmos. A matriz resultante completa, está representada abaixo.

0 1 1 1 0

1 -2 -1 -2 1

1 -1 0 -1 1

1 -2 -1 -2 1

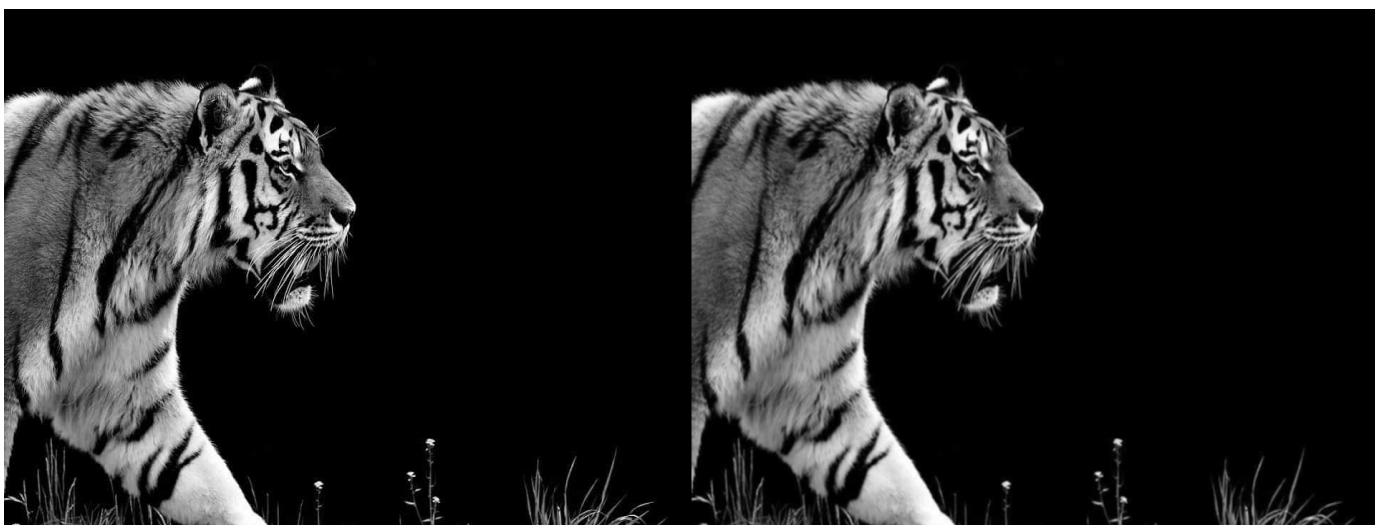
0 1 1 1 0

6. Qual o resultado esperado de uma operação que calcula a diferença absoluta entre uma imagem (em tons de cinza com tons entre 0 e 255) e essa mesma imagem filtrada por um filtro passa-baixa (um filtro Box, por exemplo)? Que tipo de filtro alcançaria esse mesmo resultado? Explique em detalhes. Ou seja, $I_1 = I - h * I$, onde I é a imagem original, h o filtro passa-baixa (como o Box) e $*$ a operação de convolução.

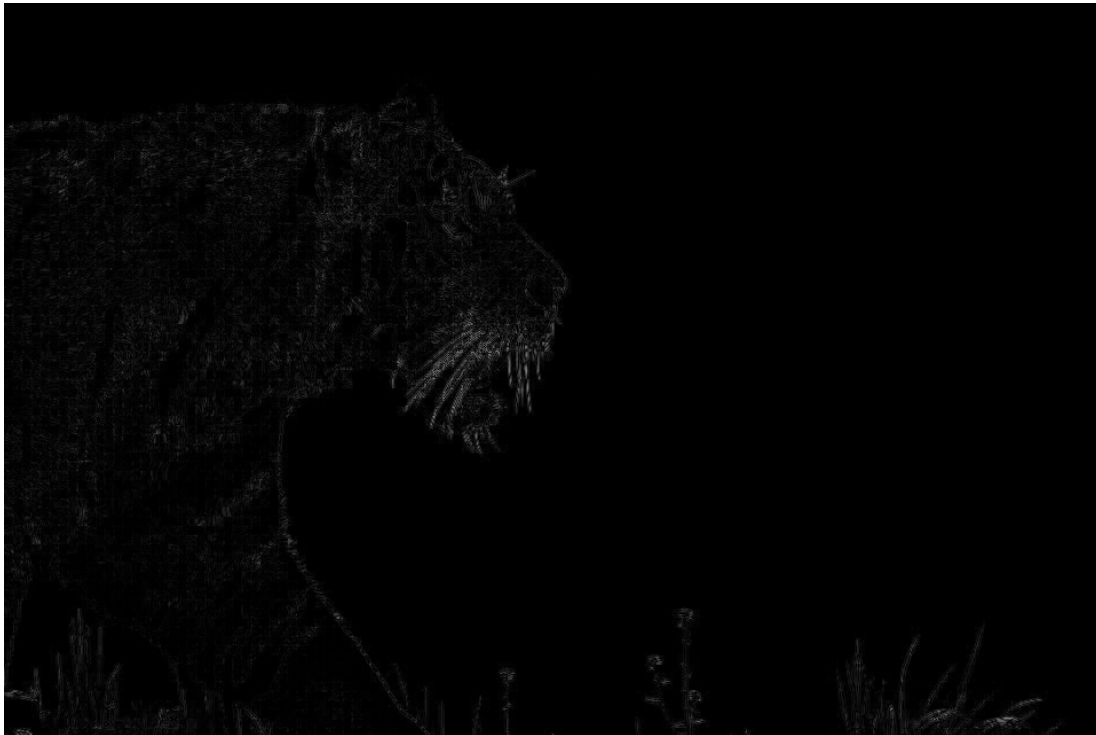
O filtro passa-baixa, como o filtro Box, realiza uma suavização da imagem, tornando-a mais borrada, onde há altas frequências, atenuando-a. Já as de baixa frequência, há pouca mudança após a aplicação do filtro box. Logo, fica mais nítido os efeitos da máscara nas bordas da imagem. Ao subtrair esse resultado da imagem original, iremos remover os pixels em que não houve mudanças significativas após a aplicação do filtro, ou seja os de baixa frequência, passando a ter um valor próximo de 0 (preto) nesses pixels, assim ficaremos com uma figura que contém as bordas da imagem, onde por terem uma alta frequência, houve uma atenuação e borramento, e graças a isso possuem uma diferença de valores maior em relação a imagem original, logo ao fazer a subtração, esses pixels ainda permanecerão no final da operação e os outros pixels de menor frequência, ficarão próximos do preto. Isso pode ser observado nas seguintes duas imagens.

Imagem

Filtro box $\rightarrow h * I$



Resultado $\rightarrow I - h * I$



Um filtro que conseguiria um resultado parecido com a operação dessa questão, seria o filtro DoG, como podemos ver abaixo na imagem, em que foi aplicado no Colab, um filtro DoG com essas configurações e código:

```
gaussian1 = cv2.GaussianBlur(img, (5, 5), 0)
gaussian2 = cv2.GaussianBlur(img, (9, 9), 0)
dog = cv2.absdiff(gaussian1, gaussian2)
```

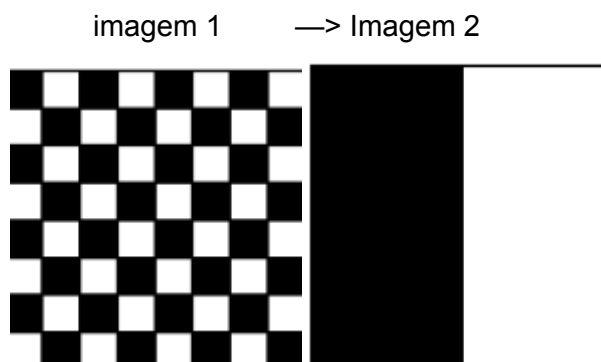


7. As imagens abaixo (ambas em preto e branco apenas) são bastante diferentes, mas seus histogramas são absolutamente iguais. Suponha que cada imagem é embaçada com um filtro da média 3x3. Os histogramas das imagens embaçadas ainda serão iguais? Justifique e esboce como devem ser o(s) histograma(s). Obs: As bordas não fazem parte das figuras; são apenas para melhor visualização.

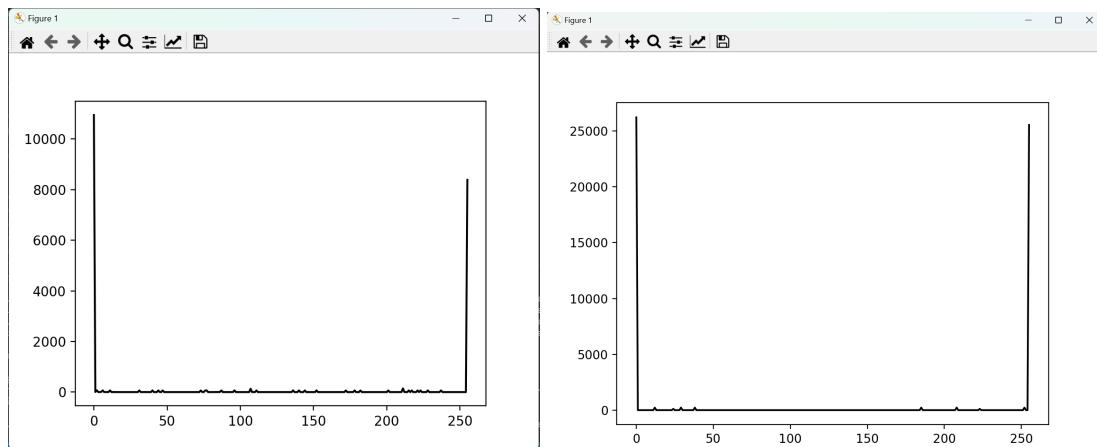
O filtro de média 3x3 ao ser utilizado, acarretará em uma suavização que resulta na diminuição de detalhes e texturas das imagens, realizando a média dos valores de intensidade dos pixels em uma determinada região, o que resulta em transições mais suaves nas regiões de alta frequência. Logo, mesmo que os histogramas das duas figuras iniciais sejam iguais, ao aplicar o filtro de média 3x3 nelas, os histogramas das representações borradas não serão idênticos, apesar de continuarem bastante semelhantes, onde poderemos ver por exemplo os mesmo picos de alta intensidade, antes da aplicação do filtro, depois de aplicá-lo.

Dessa maneira, os histogramas das figuras borradas não serão iguais, mas as figuras serão visualmente mais suaves e terão menos detalhes em relação às representações originais.

Abaixo estão as duas imagens originais, seguidas de seus respectivos histogramas.

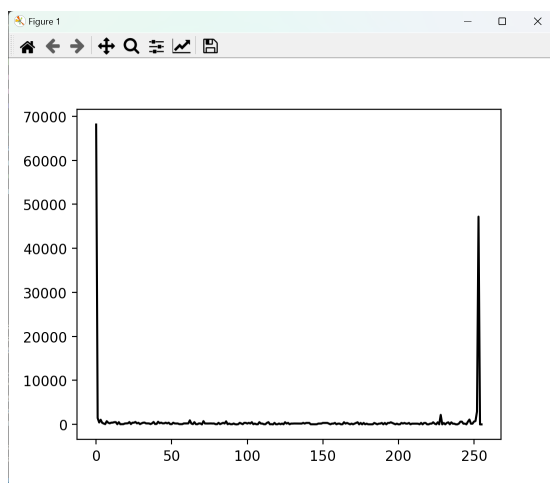


Histograma da imagem 1 —> Histograma da imagem 2

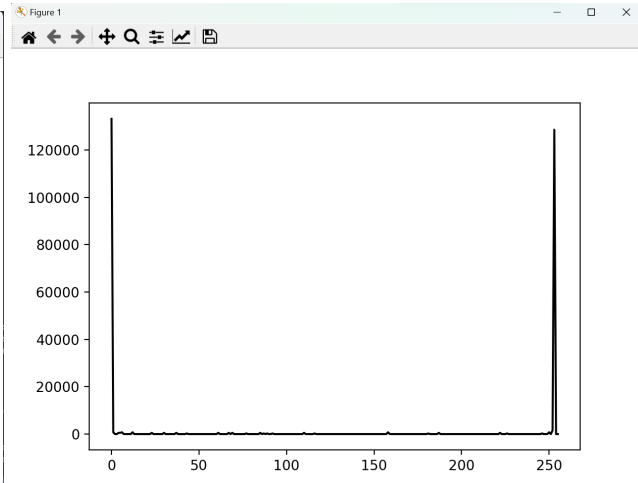


Aplicando os filtro média 3x3, temos os seguintes histogramas.

Histograma da imagem 1 com filtro



Histograma da imagem 2 com filtro



A aplicação do filtro de média 3x3 nas duas imagens afeta os histogramas, levemente dando pequenas variações nos formatos das baixas frequências e mantendo os picos de intensidade nas cores preto e branco que existiam anteriormente. Na imagem 1, podemos ver uma maior variação nos valores de transição que seriam referentes aos cinzas nessas imagens. Na imagem 2, não há muita variação nessa região, o que demonstra a leve diferença em seus histogramas após a aplicação do filtro.

8. Um filtro com muitas aplicações é o de Diferença de Gaussianas. Sejam $G_1(1)$ e $G_2(2)$ duas funções Gaussianas diferentes, o filtro DoG (Difference of Gaussians) é dado por: $DoG = G_1(1) - G_2(2)$

O filtro Gaussiano sozinho tem a função de um filtro passa baixa e, como tal, embaça uma imagem. Nessa combinação de filtros Gaussianos diferentes, como você imagina que será a aplicação do filtro DoG em uma imagem?

O filtro DoG é normalmente usado para detecção de bordas, e é obtido pela diferença entre duas funções Gaussianas com diferentes desvios padrão (σ). Ao aplicar o filtro DoG, irá borrar as bordas da imagem e suavizar essas áreas de maior frequência. Inicialmente, o filtro Gaussiano com o desvio padrão σ_1 é aplicado à imagem. Em seguida, o filtro Gaussiano com o desvio padrão σ_2 , é aplicado à imagem. Esse filtro também suaviza a imagem, mas de uma maneira mais intensa devido ao desvio padrão maior. Ao subtrair o DoG com desvio padrão σ_2 do com σ_1 , iremos tornar os pixels de baixa frequência mais próximos do preto, ficando com as bordas onde os pixels da imagem sofreram uma atenuação. Essa diferença de imagens realça as mudanças bruscas de intensidade, que são indicativas das bordas na imagem original. O filtro DoG é eficaz na detecção de bordas, pois realça as transições de intensidade, enquanto suaviza outras áreas de intensidade uniforme. As bordas são destacadas como áreas de alta resposta, enquanto regiões sem bordas tendem a ter uma resposta próxima a zero.

Portanto, a aplicação do filtro DoG em uma imagem resulta em uma imagem filtrada onde as bordas são realçadas, tornando-o útil para tarefas de detecção de bordas.

9. Explique porque as operações de erosão e dilatação (com um mesmo elemento estruturante) não são necessariamente operações inversíveis.

As operações de Erosão e Dilatação, se referem respectivamente à diminuição dos pixels brancos e à expansão deles, no entanto, não são operações inversas ou inversíveis, pois não necessariamente o inverso de uma erosão irá resultar numa dilatação, pode sim ocorrer, mas não por definição, e sim dependendo de um contexto específico e das características da imagem em questão. Numa Erosão com um determinado elemento, os pixels brancos da imagem, que não casem completamente com os pixels ativos do elemento, virarão pretos e apenas os que casam completamente, serão pintados de branco no elemento central do elemento estruturante. Já na dilatação qualquer pixel da imagem que case com os pixels brancos do elemento estruturante, serão pintados de branco. Ao aplicar a erosão seguida da dilatação com o mesmo elemento estruturante, é obtida a operação conhecida como abertura. A abertura é comumente usada para remover ruídos e pequenos detalhes, além de separar objetos que estejam muito próximos. No entanto, é importante ressaltar que a aplicação da dilatação seguida da erosão com

o mesmo elemento estruturante não garante a reversibilidade da operação. Isso ocorre porque a erosão pode remover detalhes e modificar a forma dos objetos. Essas alterações podem resultar em perda de informação que não pode ser totalmente recuperada pela dilatação subsequente. Por isso, as operações de erosão e dilatação não são necessariamente reversíveis.

Atividade Prática

Considere a imagem Flor_Joaninha.jpg. Essa imagem apresenta um efeito chamado de Low-Depth of Field (baixo campo de profundidade), onde um objeto em foco aparece em primeiro plano com o fundo da imagem desfocado. Tente segmentar essa imagem automaticamente de forma a preservar apenas os objetos em primeiro plano.

O código está representado na sequência abaixo, junto com os comentários em verde, e as fotos dos resultados de partes do código rodado por etapas.

1)

```
import matplotlib.pyplot as plt
import cv2
from google.colab.patches import cv2_imshow
import numpy as np
%matplotlib inline

!git clone https://github.com/MZMLPE/Imagens

# Caminho da imagem
path = "/content/Imagens/Flor_Joaninha.jpg"
# Lendo a imagem
img = cv2.imread(path)
# Mostra a imagem no colab
cv2_imshow(img)
```



2)

```
# Transforma a imagem em escala de cinza  
img_gray = cv2.imread(path, 0)  
cv2_imshow(img_gray)
```



3)

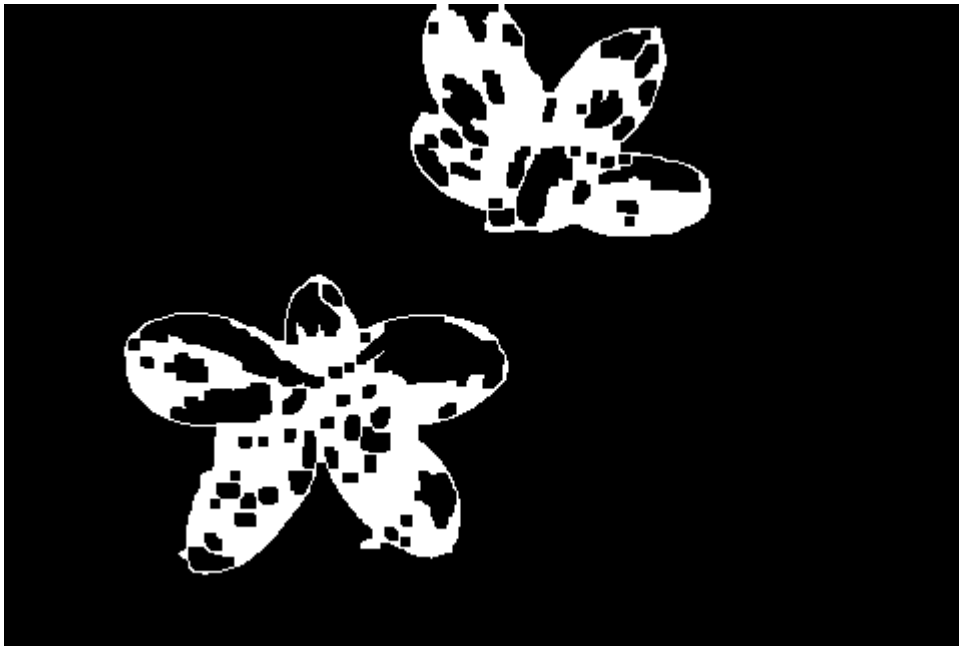
```
# O algoritmo de Canny foi utilizado para detectar as bordas da imagem
```

```
bordas = cv2.Canny(img, 20, 290)
cv2_imshow(bordas)
```



4)

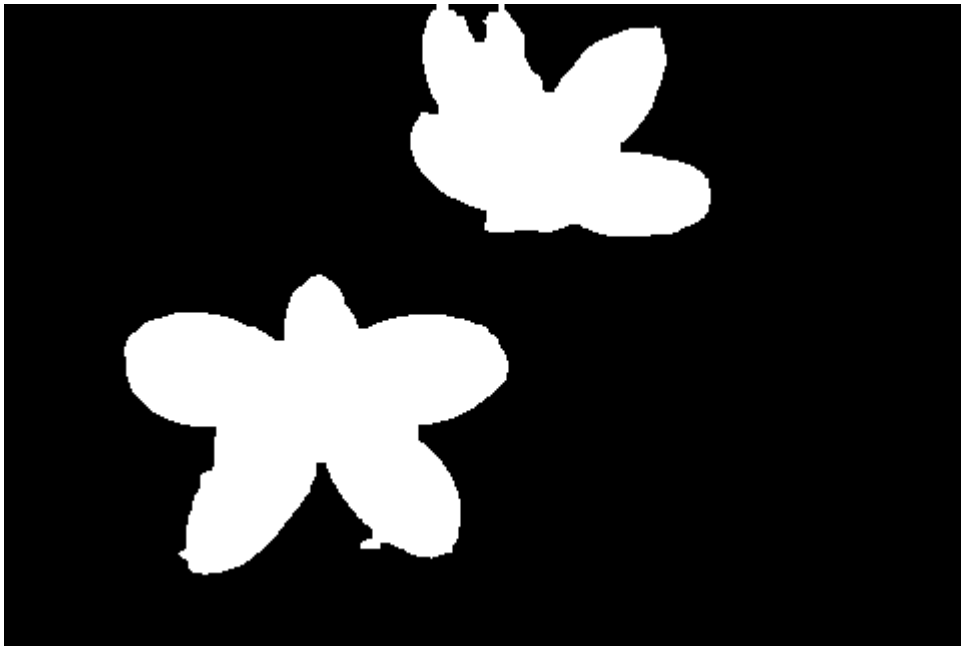
```
# Definindo um elemento estruturante com dimensão 5
elemento = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5))
# Realiza um fechamento morfológico, fazendo uma erosão
# seguida da dilatação da imagem. O que elimina ruídos
# e depois reforça os pixels de contorno
fechamento = cv2.morphologyEx(bordas, cv2.MORPH_CLOSE, elemento)
cv2_imshow(fechamento)
```



5)

```
# Encontrando apenas contornos externos dos objetos para preenchimento,
desconsiderando os internos
contours, _ = cv2.findContours(fechamento, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

# Preenchendo o contorno e criando uma máscara desses pixels
mask = cv2.drawContours(np.zeros_like(img), contours, -1, (255, 255,
255), thickness=cv2.FILLED)
cv2.imshow(mask)
```



6)

```
# Remove os pixels preto que aparecem na máscara e deixa apenas  
# os contornos preenchidos anteriormente, mostrando apenas  
# os objetos em primeiro plano  
resultado = cv2.bitwise_and(img, mask)  
  
cv2_imshow(resultado)
```

