

Programação orientada a objetos

Thresholding e filtros

Conceitos importantes

- **Thresholding**

Thresholding é o método mais simples, em Processamento Digital de Imagens, de segmentação de imagens. Ao receber como entrada uma imagem em tons de cinza (*grayscale*) e um valor inteiro, T , que representa o limiar (*threshold*), é gerada, como resultado, uma imagem binária, de acordo com a seguinte regra:

$$I_{out}(x, y) = \begin{cases} 255 & , I_{in}(x, y) > T \\ 0 & , I_{in}(x, y) \leq T \end{cases} \quad (1)$$

em que $I_{out}(x, y)$ representa o *pixel* na posição (x, y) da imagem binária resultante, $I_{in}(x, y)$ é o *pixel* correspondente da imagem *grayscale* de entrada, e T é o valor de *threshold*.

- **Exemplo**



Original image



$T = 150$



$T = 200$

- **Limiarização global simples**

A aplicação de *thresholding* pode ser bastante útil para segmentar objetos em imagens. No entanto, a definição do valor mais adequado de T pode ser subjetiva e interferir na imagem resultante. Diferentes métodos foram propostos na literatura, buscando otimizar a definição desse valor, baseado no histograma da imagem, como o método do vale, a Limiarização Global Simples proposta por Gonzalez & Woods, e a Limiarização Global Ótima utilizando o método de Otsu. Para fins da execução

deste trabalho, será explicado apenas o método de Limiarização Global Simples; para conhecimento dos demais métodos, recomenda-se a leitura do material auxiliar.

Quando as distribuições de intensidade dos *pixels* de fundo e dos objetos são suficientemente diferentes, é possível utilizar um único limiar (global) aplicável a toda imagem. O método iterativo proposto por Gonzalez & Woods pode ser utilizado nesses casos, e funciona da seguinte forma:

1. Selecionar uma estimativa inicial para o limiar global, T ;
2. Segmentar a imagem usando T , de acordo com a Equação 1 apresentada anteriormente. Isso dará origem a dois grupos de *pixels*: G_1 , composto por todos os *pixels* com valores de intensidade maiores que T , e G_2 , composto por *pixels* com valores menores ou iguais a T ;
3. Calcular os valores de intensidade média, m_1 e m_2 para os *pixels* em G_1 e G_2 , respectivamente;
4. Calcular um novo valor de limiar da forma:

$$T = \frac{1}{2}(m_1 + m_2) \quad (2)$$

5. Repita as etapas 2 a 4 até que a diferença entre os valores de T em iterações sucessivas seja menor que o parâmetro ΔT , predefinido.

Este algoritmo simples funciona bem em situações em que há um vale razoavelmente claro entre os modos de histograma relacionados aos objetos e ao fundo. O parâmetro ΔT é usado para controlar o número de iterações em situações nas quais a velocidade é uma questão importante. Em geral, quanto maior for o ΔT , menor é o número de iterações que o algoritmo executará. O limiar inicial deve ser maior do que o mínimo e menor do que o máximo nível de intensidade na imagem. A intensidade média da imagem é uma boa escolha inicial para T .

● Filtros

Filtros da média e da mediana são técnicas amplamente utilizadas no processamento de imagens para suavizar as características e reduzir ruídos das imagens. Esses filtros são aplicados a uma matriz de *pixels* que representa a imagem, com o objetivo de substituir o valor de cada *pixel* pela média ou mediana dos valores de sua vizinhança.

O filtro da média calcula a média aritmética dos valores dos *pixels* em uma determinada vizinhança e atribui esse valor ao *pixel* central. Ao fazer isso para todos os *pixels* da imagem, o filtro da média suaviza a imagem, reduzindo variações abruptas nos níveis de intensidade.

Já o filtro da mediana substitui o valor de cada *pixel* pelo valor mediano dos *pixels* em sua vizinhança. O valor mediano é o valor central quando os valores da vizinhança são ordenados em ordem crescente ou decrescente. Esse filtro é particularmente útil para remover ruídos impulsivos (como "salt and pepper") da imagem, pois o valor mediano é menos afetado por valores extremos.

Todas as operações realizadas devem gerar uma nova imagem, sempre mantendo a imagem original, utilizada como entrada para o método, imutáveis.

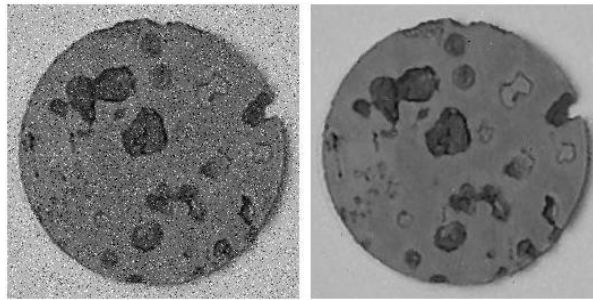
É importante atentar-se às regiões das bordas da imagem, uma vez que os *pixels* das extremidades possuem um número reduzido de vizinhos. Nesse caso, duas principais abordagens podem ser adotadas: *zero padding*, que consiste na criação de uma borda na imagem preenchida com zeros, de forma a ajustar o número de *pixels* necessário para a execução da filtragem; ou *same padding*, que realiza o mesmo procedimento, com a diferença de que ao invés de preencher novos *pixels* com zero, esta abordagem replica os valores dos *pixels* da extremidade para usá-los como borda. A seguir, são apresentados exemplos das duas abordagens.

65	65	12	98	98	0	0	0	0	0
65	65	12	98	98	0	65	12	98	0
56	56	32	98	98	0	56	32	98	0
78	78	48	63	63	0	78	48	63	0
78	78	48	63	63	0	0	0	0	0
(a) <i>same padding</i>					(b) <i>zero padding</i>				

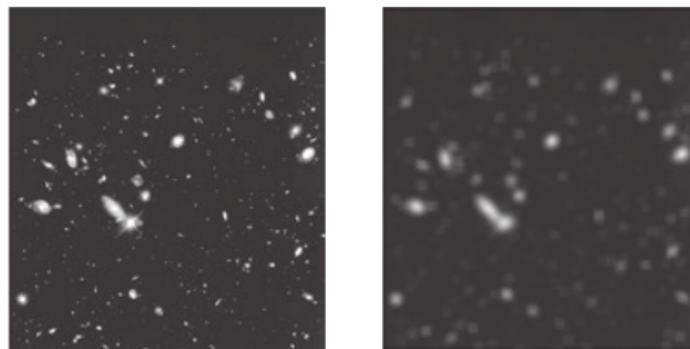
Originalmente, a imagem corresponde à região branca das imagens. As bordas criadas pelas abordagens *same padding* e *zero padding* são representadas pela região verde nas duas imagens.

Ambos os filtros podem ser implementados com diferentes tamanhos de vizinhança, preferencialmente ímpares, em que um tamanho maior considera uma área maior ao redor do *pixel* central. No entanto, é importante encontrar um equilíbrio, pois aumentar muito o tamanho da vizinhança pode levar à perda de detalhes importantes da imagem.

- **Exemplos**



(a) imagem original (b) após filtro da mediana



(a) imagem original (b) após filtro da média

Tarefas

Você deve organizar seu código como uma classe, *Image*, que armazenará, como atributos, as informações pertinentes à imagem: o número mágico utilizado (P2 ou P5), as dimensões, o valor de *maxval*, e os *pixels* propriamente ditos. O histograma da imagem deve ser armazenado também como um atributo da classe. Este histograma deve ser representado como uma lista com 256 posições, em que cada uma delas representa a intensidade do *pixel*, e o valor armazenado naquela posição representa a quantidade de *pixels* daquela intensidade, como apresentado a seguir:



em que 0, 1, 2, ..., 254, 255 são os índices das posições da lista, e *a*, *b*, *c*, ..., *y*, *z* representam as quantidades de *pixels* com intensidades 0, 1, 2, ..., 254, 255, respectivamente. Cada atributo deve possuir seu respectivo método de leitura e, quando pertinente, o método de escrita.

Você deve implementar, entre outros, os seguintes métodos:

- **thresholding(t)** : que realiza o *thresholding* na imagem, considerando o valor t e gera uma imagem binária. Este método deve retornar um novo objeto do tipo *Image*, com todas as informações pertinentes, e a imagem resultante deve ser salva em diretório especificado (ver seção **Entrada**). Por padrão, o valor de t deve ser assumido como 127;
- **sgt(dt)** : que calcula o valor do *threshold* global, de acordo com o método de Limiarização Global Simples apresentado anteriormente. O parâmetro dt representa a variação do *threshold* a ser considerada como critério de parada do método, de acordo com a descrição, e, por padrão, assume o valor 1. O método deve armazenar o valor T calculado como um atributo da classe e retornar um novo objeto do tipo *Image* contendo a imagem binarizada utilizando o valor T como *threshold* e todas as informações pertinentes. A imagem resultante deve ser salva em diretório especificado (ver seção **Entrada**);
- **mean(k)** : que aplica o filtro da média na imagem, considerando uma vizinhança quadrada de tamanho $k \times k$. Por padrão, deve ser considerado $k = 3$, caso nenhum valor seja passado como entrada. Este método deve retornar um novo objeto do tipo *Image*, com todas as informações pertinentes, e a imagem resultante deve ser salva em diretório especificado (ver seção **Entrada**);
- **median(k)** : que aplica o filtro da mediana na imagem, considerando uma vizinhança quadrada de tamanho $k \times k$. Por padrão, deve ser considerado $k = 3$, caso nenhum valor seja passado como entrada. Este método deve retornar um novo objeto do tipo *Image*, com todas as informações pertinentes, e a imagem resultante deve ser salva em diretório especificado (ver seção **Entrada**).

Entrada

A passagem dos parâmetros de entrada do programa deve ser feita por linha de comando, seguindo o padrão especificado abaixo:

```
python <nome_programa>.py --imgpath <caminho_imagem_entrada> --op
<operacao> --<param> <valor_parametro> [--outputpath <caminho_imagem_saida>]
```

A seguir, são descritos os parâmetros:

- **--imgpath** : consiste no caminho para a imagem, incluindo o nome;
- **--op** : consiste na operação a ser realizada, em que os valores possíveis são *thresholding*, *sgt*, *mean* ou *median*;
- **--<param>** : deve ser substituído pelo nome adequado à operação selecionada na opção anterior, podendo ser $-t$, $-dt$ ou $-k$, para o *thresholding*, *sgt* ou *mean/median*, respectivamente. Por exemplo, se foi selecionado *--op sgt*, então *--<param>* deve

ser substituído por `--dt`, e utilizado um valor subsequentemente, da forma `--dt 1`;

- **--outputpath** : consiste no diretório em que as imagens resultantes serão salvas; note que, diferente de `--imgpath`, este parâmetro **não** inclui o nome da imagem resultante, pois este deve estar definido no código. Por padrão, caso esta opção não seja utilizada pelo usuário, deve-se considerar o diretório corrente.

Saída

Seu programa deve salvar, quando pertinente, as imagens de saída no diretório especificado com o parâmetro `--outputpath`. É recomendado que o nome da imagem seja igual ao nome do método que a gerou, por exemplo, *mean.pgm* quando a imagem resultante for gerada pela aplicação do filtro da média. Além disso, como saída na tela, seu programa deve exibir as seguintes informações da imagem: número mágico, dimensões, *maxval*, a média e a mediana dos *pixels* da imagem, e o valor *T* obtido determinado pelo método *sgt()*. O padrão a ser seguido deve ser:

```
magic_number <value>
dimensions (x, y)
maxval <value>
mean <value>
median <value>
T <value>
```

Observações

- O código deve estar num arquivo Python simples (não notebook);
- não é permitido o uso de pacotes de manipulação de imagens;
- para a leitura dos parâmetros de entrada, é permitida a utilização do pacote *argparse*;
- lembre-se de organizar o seu código de forma apropriada, levando em conta as boas práticas de programação.

Material auxiliar

- https://medium.com/@Orca_Thunder/image-padding-techniques-zero-padding-part-1-669de127ba59
- https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html
- <https://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm>
- https://en.wikipedia.org/wiki/Median_filter
- https://sde.uoc.ac.in/sites/default/files/sde_videos/Digital%20Image%20Processing%203rd%20ed.%20-%20R.%20Gonzalez,%20R.%20Woods-ilovepdf-compressed.pdf