scc0251/5830— Prof. Moacir Ponti

PAE/monitores: Fernando Pereira dos Santos, Eduardo Carlos Souza

Trabalho 3: filtragem 1D

Desenvolva seu código sem olhar o de colegas. Plágio não será tolerado.

Nesse trabalho você deverá implementar dois métodos distintos de filtragem 1D, considerando a imagem como um vetor unidimensional.

Você deverá implementar usando `python 3` com as bibliotecas `numpy` e `imageio`.

Leia com calma as instruções de cada etapa.

Seu programa deverá permitir receber o nome de uma imagem, a partir da qual você irá aplicar um dos dois métodos de filtragem 1D escolhidos (arbitrário ou função gaussiana) e, posteriormente, fornecer a imagem filtrada como saída.

1. **Receber via teclado**:
   a) nome da imagem para filtragem (I);
   b) opção de escolha do filtro (1 – arbitrário, 2 – função Gaussiana);
   c) tamanho do filtro n;
   d) parâmetro(s) que define(m) os pesos do filtro:
      – para o método 1, uma sequência de n pesos, $w_{A1}, w_{A2}, \ldots, w_{An}$;
      – para o método 2, um único valor que representa o desvio padrão da distribuição, σ;
   e) domínio da filtragem (1 – espacial, 2 — frequência).

Imagem vetor

Após a leitura da imagem, faça ajustes para que a matriz recebida (imagem) I, tenha o formato de um único vetor unidimensional, gerando $\mathbf{I}$, sendo que você deverá posicionar as linhas de valores em sequência. Por exemplo dada uma imagem de entrada no formato:

$$I = \begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix},$$

Devemos obter:

$$\mathbf{I} = \begin{bmatrix} 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \end{bmatrix}$$

e utilizar esse vetor para realizar a filtragem unidimensional. Após o processamento, faça o processo inverso para formar a imagem de saída. O processo de normalização dos valores (0 - 255) e a conversão para números inteiros deverá ocorrer somente na comparação do seu resultado, portanto mantenha os resultados parciais com casas decimais.

Escolha do Filtro

Opção 1: Filtro Arbitrário

Você deverá implementar um filtro unidimensional arbitrário $\mathbf{w}_A$ de tamanho $1 \times n$, formado por uma máscara de pesos (com valores reais) fornecida pelo usuário (via teclado). O valor central do filtro define a respectiva posição de saída. Por exemplo, seja $n = 5$ e a imagem e filtros dados por:

$$\begin{aligned} \mathbf{I} &= \begin{bmatrix} 10 & 11 & \textit{12} & 13 & 14 & 15 & 16 & 17 & 18 \end{bmatrix}, \\ \mathbf{w}_A &= \begin{bmatrix} 0.4 & 0.2 & 0.1 & 0 & 0.3 \end{bmatrix} \end{aligned}$$

Então podemos calcular a posição 2 da imagem processada (relativa ao valor 12), da seguinte forma:

$$\lfloor (10 \cdot 0.4) + (11 \cdot 0.2) + (12 \cdot 0.1) + (13 \cdot 0) + (14 \cdot 0.3) \rfloor = \lfloor 11.6 \rfloor = 11$$

Assim, já temos o valor da posição 2 e a imagem processada (em formato vetorial) então seria:

$$\hat{\mathbf{I}} = \begin{bmatrix} ? & ? & 11 & ? & ? & ? & ? & ? & ? \end{bmatrix}.$$

Os outros elementos, ainda não processados estão indicados com o ponto de interrogação (?).

Nos casos em que o filtro possuir tamanho par, o valor central será definido por $(n/2) - 1$. Por exemplo, em um filtro de tamanho 4, o segundo valor será a posição central.

Opção 2: Filtro Gaussiano

Para o filtro Gaussiano $\mathbf{w}_G$ será informado o tamanho do filtro e o valor do desvio padrão da distribuição Gaussiana (σ). Neste filtro, para cada posição do vetor aplique a equação:

$$G_{1D}(x, \sigma) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp\left(\frac{-\left(x^2\right)}{2\sigma^2} \right)$$

Importante! Essa equação considera que o filtro está centrado na origem (0) e que os valores de x para a esquerda são negativos, e para a direita são positivos. Ex. se um filtro tiver tamanho $n = 5$, então você deverá computar o filtro para os diversos valores de $x \in \mathbf{Z}$ centrados em 0. Nesse caso, $x = -2, -1, 0, 1, 2$, formando o filtro com 5 pesos.

Além disso, você deve normalizar o filtro Gaussiano de forma a garantir que os pesos tenham soma unitária, i.e., $\sum \mathbf{w}_G = 1$

Domínio de filtragem

Tipo 1: Domínio Espacial

Realiza a convolução no domínio espacial, deslocando o filtro ao longo da imagem e computando a operação de convolução. Nesse caso você deverá considerar que a imagem é um vetor circular (*image wrap*), permitindo calcular todos os pixels da imagem processada.

Pegando como exemplo o caso apresentado no filtro arbitrário, para calcular a primeira posição do vetor (posição 0), faremos a convolução com a seguinte configuração (de posição relativa):

$$\begin{array}{rccccccccc} \text{posicoes do vetor} & 7 & 8 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \text{valores} & [017 & 018 & 010 & 011 & 012 & 013 & 014 & 015 & 016] \\ \mathbf{w}_A & [0.4 & 0.2 & 0.1 & 0.0 & 0.3] & & & & \end{array}$$

Tipo 2: Domínio da Frequência

De acordo com o teorema da Convolução, temos que a convolução no domínio espacial é equivalente ao produto no domínio da frequência:

$$w(x) * f(x) \equiv \mathfrak{F}^{-1}(W(u) \cdot F(u)),$$

sendo que $W(u)$ e $F(u)$ são as transformadas de Fourier do filtro e da imagem, respectivamente, e $\mathfrak{F}^{-1}()$ é a transformada inversa de Fourier.

Assim, você deverá implementar suas próprias funções para a transformada discreta de Fourier, e sua inversa, computar $W(u)$ e $F(u)$, realizar o produto ponto-a-ponto no domínio da frequência, e depois obter a inversa do vetor resultante.

Observação 1: Para realizar o produto ponto-a-ponto, preencha o restante dos valores do filtro com zero antes de realizar a transformada.

Observação 2: É estritamente proibido fazer uso das funções de Fourier contidas na biblioteca numpy. É obrigatório o desenvolvimento de seu próprio método.

Observação 3: Para considerar somente a parte real dos números complexos, utilize a função np.real() da biblioteca numpy.

Comparação com imagem de referência

Após gerar a imagem de saída $\hat{I}$, comparar com a imagem original de entrada I utilizando o RMSE:

$$RMSE = \sqrt{\frac{1}{NM}\sum_i \sum_j (I(i,j) - \hat{I}(i,j))^2},$$

onde $N \times M$ é a resolução das imagens sendo comparadas.

Entrada e saída

Exemplo de entrada 1: imagem de entrada, filtro arbitrário (1), tamanho $n = 5$, pesos (número de pesos é igual ao tamanho), conforme abaixo, e tipo (domínio espacial):

```
cat.png
1
5
0.4 0.2 0.1 0 0.3
1
```

Exemplo de entrada 2: imagem de entrada, filtro Gaussiano (2), tamanho $n = 7$, sigma 0.7 e tipo (domínio da frequência), conforme abaixo:

```
cat.png
2
7
0.7
2
```

Exemplos de saída: apenas o valor do RMSE em formato `float`

```
2.0897
```

Submissão e instruções

No sistema Run.Codes deve incluir apenas o arquivo `.py`

1. **É obrigatório comentar seu código**. Como cabeçalho insira seu nome, número USP, código da disciplina, ano/semestre e o título do trabalho. Haverá desconto na nota caso esse cabeçalho esteja faltando, além de descontos para falta de comentários.
2. **É obrigatório organizar seu código em funções**. Utilize uma função para gerar/montar o filtro arbitrário e outra para o filtro gaussiano que retorne um vetor com os filtros. Utilize uma função que realize a convolução de um filtro com uma imagem e retorne a imagem processada. Outras auxiliares também podem ser criadas.