

# Exercícios Filtragem Espacial

1. Siga os passos para a implementação de um filtro de suavização de imagens e avaliação do resultado.
  - a. Implemente o Filtro Gaussiano utilizando o elemento estruturante 3x3 apresentado na Quadro 1. (Lembre-se que o pixel central e os seus vizinhos devem ser **multiplicados** pelas suas respectivas posições no kernel). Um exemplo de suavização é apresentado na Figura 1, onde é possível observar uma comparação entre a lena original e após o processamento.

$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$

Quadro 1. Elemento estruturante do Filtro Gaussiano

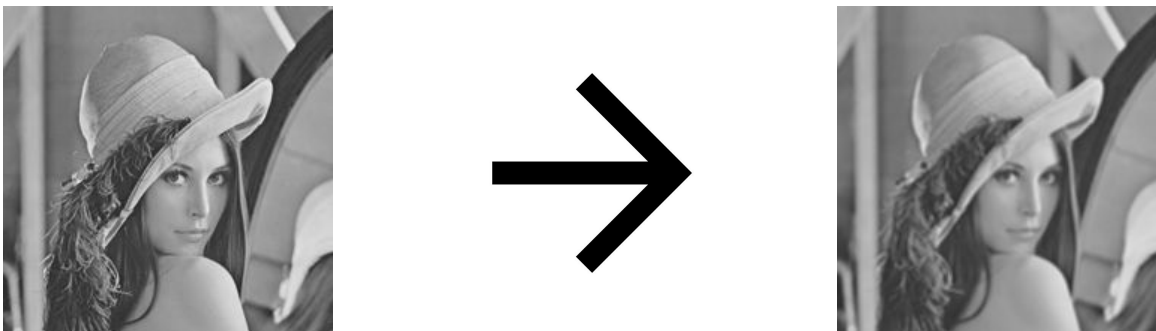


Figura 1. Operação de suavização

- b. Implemente uma função para adição de ruídos à imagem do tipo sal e pimenta (*salt and pepper*). O Quadro 2 apresenta as linhas de códigos que devem ser utilizadas para adicionar ruídos à imagem.

```
# importa a funcao random_noise do modulo skimage
from skimage.util import random_noise

# funcao que adiciona o ruído na imagem
img_ruídos = random_noise(img_original, mode='s&p', amount=0.3)
```

Quadro 2. Comando necessário para adição de ruídos na imagem

- c. Aplique o Filtro Gaussiano na imagem com ruídos. Em seguida aplique a função abaixo do módulo skimage para avaliar o PSNR (Peak signal-to-noise ratio) da imagem processada em relação à original. Faça as comparações das imagens antes e depois da aplicação do filtro.

```
# importa o modulo measure do skimage
from skimage import metrics

# mede o PSNR da imagem original em relação a suavizada
print('psnr:', metrics.peak_signal_noise_ratio(
    img_original,
    img_teste
))
```

Quadro 3. Comando para avaliar o PSNR

2. Implemente o **Filtro Laplaciano**. Para isso aplique a operação de convolução utilizando o elemento estruturante do Quadro 3.

0	- 1	0
- 1	4	- 1
0	- 1	0

Quadro 4. Exemplo de elemento estruturante do Filtro Laplaciano

3. Combine os filtros **Gaussiano** e **Laplaciano** para obter uma detecção de bordas melhor. Em ordem: aplique o gaussiano na imagem e original e em seguida aplique o laplaciano.
4. Realize novamente a implementação dos exercícios anteriores, mas utilize a função de convolução do **OpenCV** e avalie a diferença no desempenho.