Inteligência Artificial

Instrutores
Ph.D. Professor Aluisio Igor Rego Fontes

Conceitos Básicos

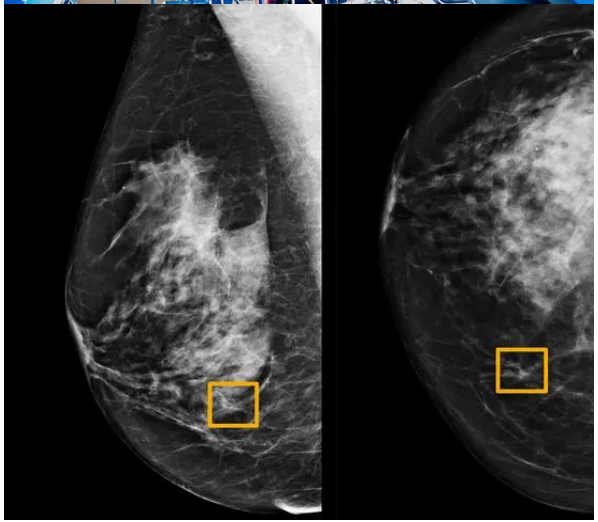
O que é processamento de imagens digitais?

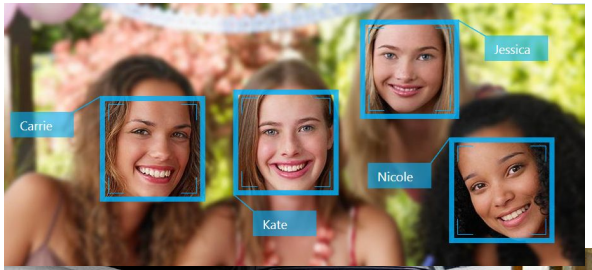
- O que é filtragem?
  - Aplicação de operações em pixels de uma imagem que vão modificar suas característica.
  - Essas aplicações, são conhecidas como filtros.
  - Bastante utilizada para **realce, suavização, remoção de ruídos, detecção de bordas**, entre outros propósitos. Ou seja, realçar alguma característica interessante da imagem ou tentar apagar algum ruído que esteja atrapalhando.
    - Ruído é caracterizado por uma falha na imagem.

Motivação - Visto em aula passada

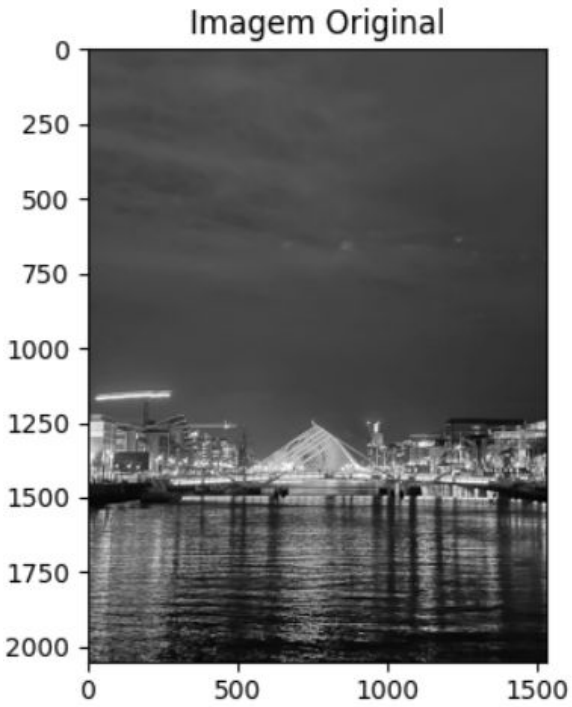
- A visão é o nosso sentido mais avançado, imagens determinam um papel extremamente importante na percepção humana.
- Extrair informações valiosas e conseguir automatizar tarefas complexas junto com a melhora de eficiência em vários polos da sociedade.
- Já usamos processamento de imagens em diversos setores, já é uma realidade, a tendência é utilizarmos ainda mais.

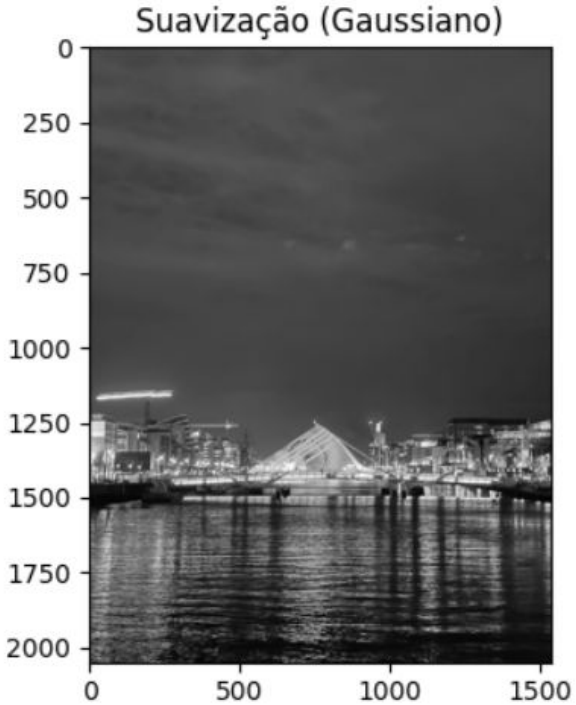
Onde é utilizado





Exemplo - Suavização



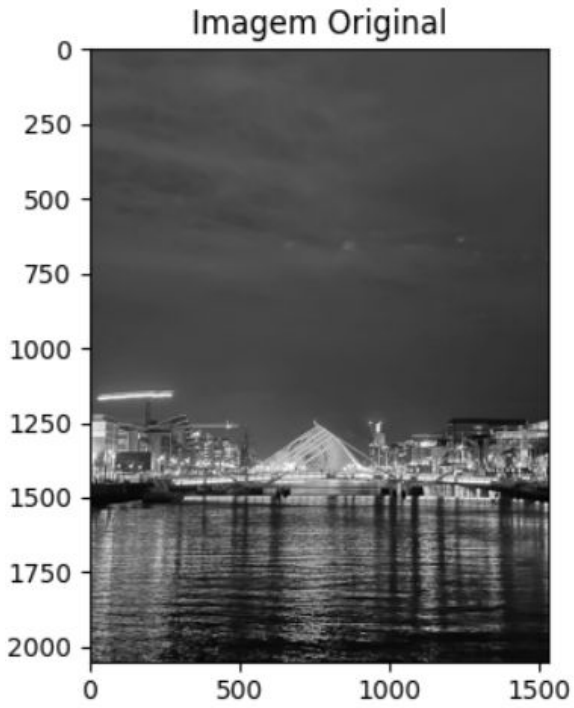


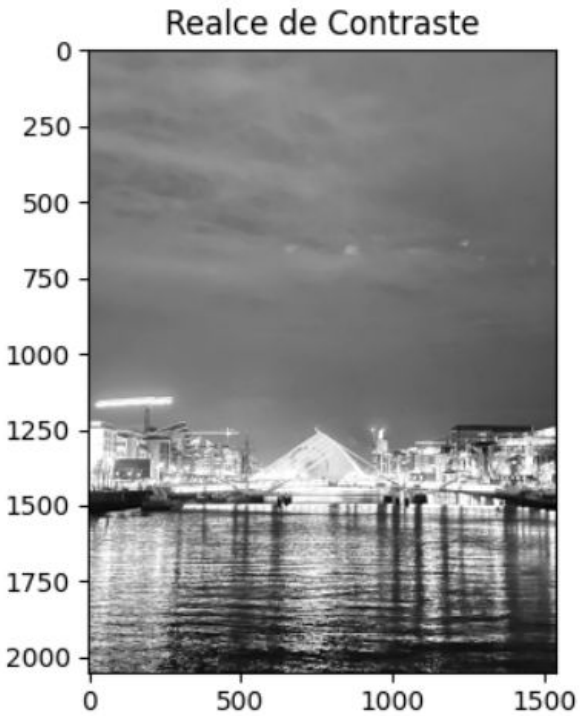
Exemplo - Suavização

- Código em python

```
image_path = 'imagem2.jpg'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

kernel_size = 5
smoothed_image = cv2.GaussianBlur(image, (kernel_size, kernel_size), 0)
```

Exemplo - Realce



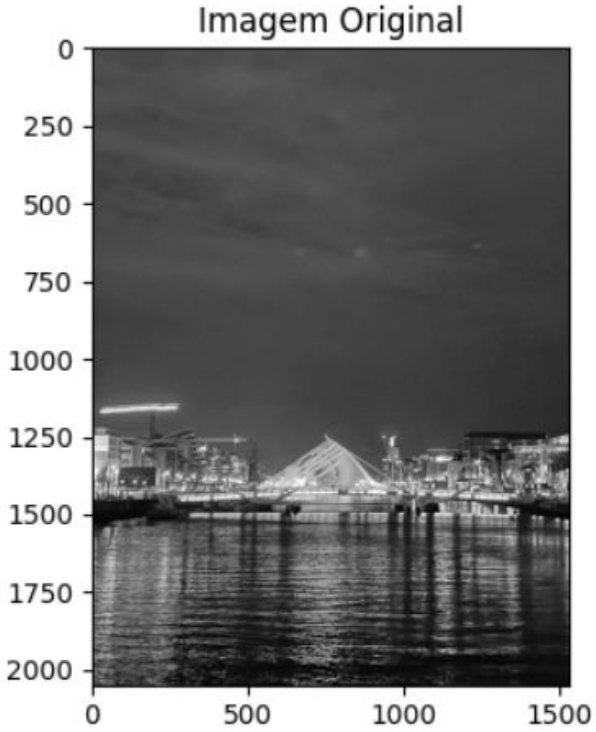


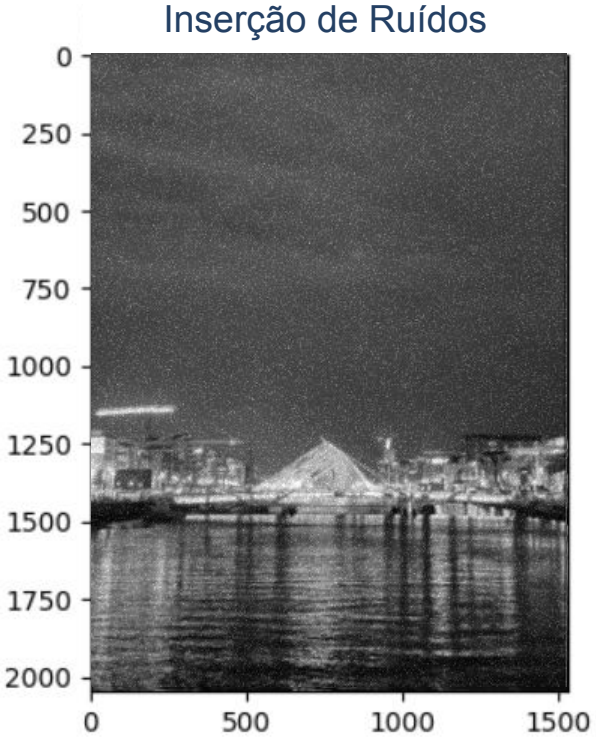
Exemplo - Realce

- Código em python
  - Para realce, temos que a transformação linear é:
    - realce(a, b) = alpha * image(a, b) + beta

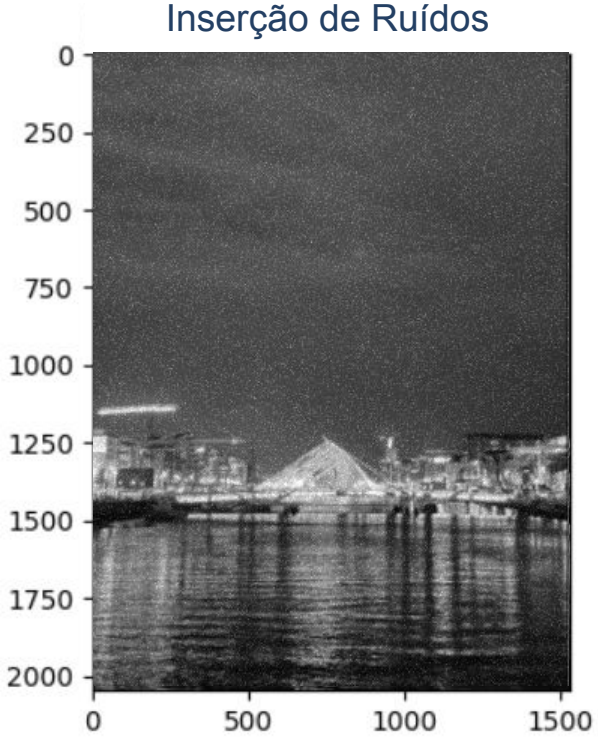
```
image_path = 'imagem2.jpg'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

alpha = 1.5
beta = 30
enhanced_image = cv2.convertScaleAbs(image, alpha=alpha, beta=beta)
```

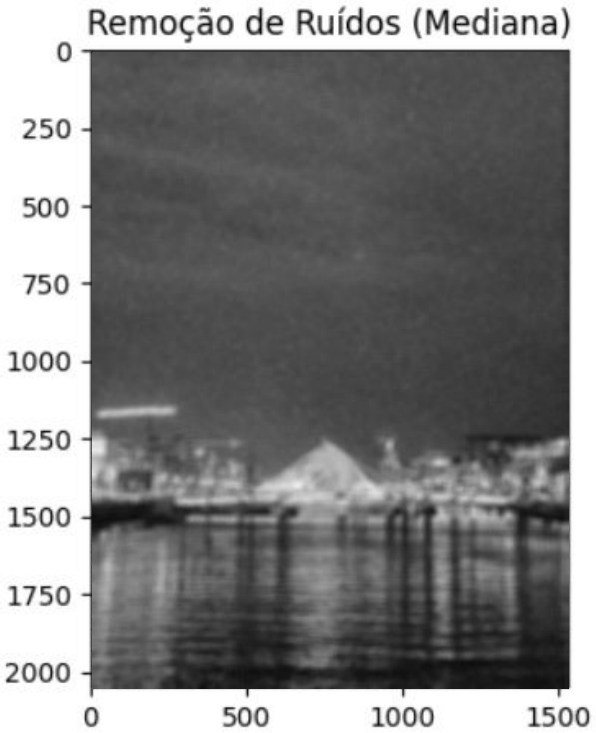
Exemplo - Remoção de ruído





Exemplo - Remoção de ruído



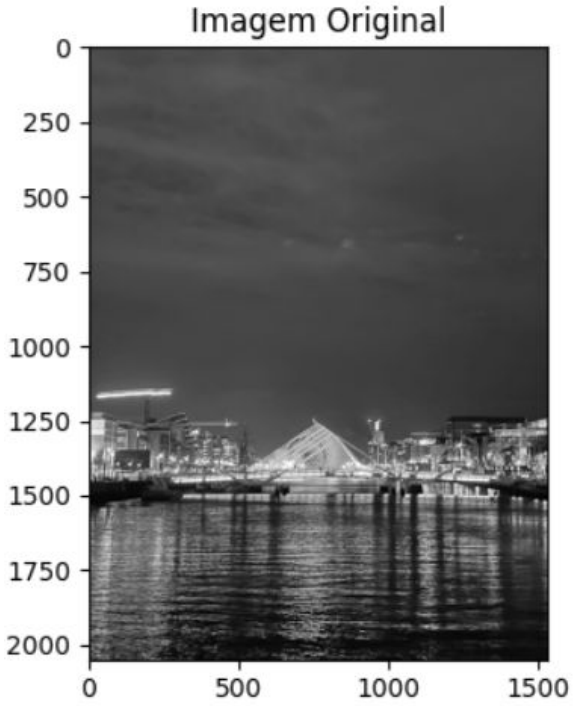


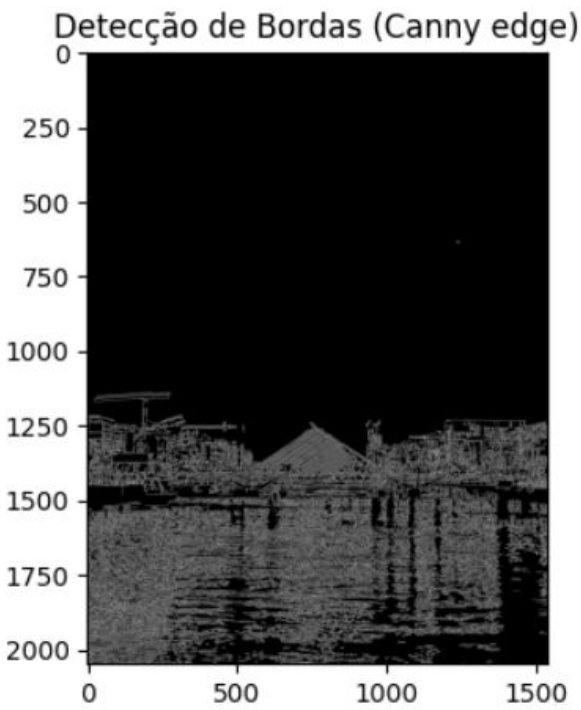
Exemplo - Realce

- Código em python

```
image_path = 'imagem2.jpg'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
median_filtered_image = cv2.medianBlur(image, 5)
```

Exemplo - Detecção de bordas



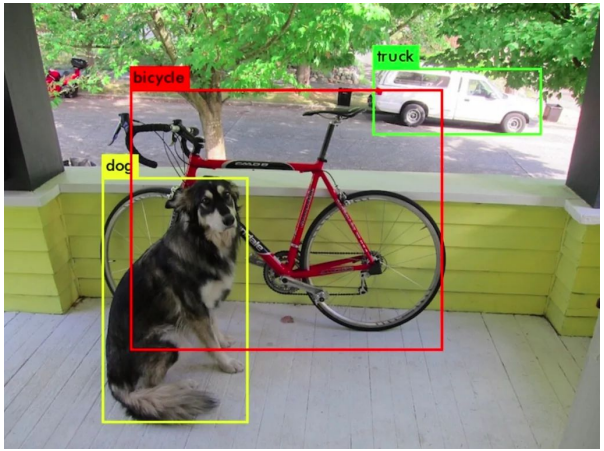


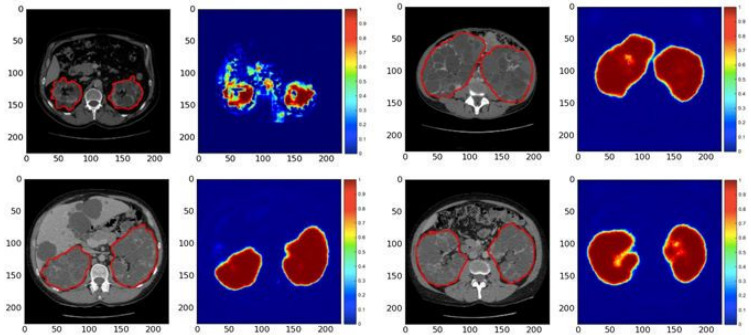
Exemplo - Detecção de bordas

- Código em Python

```
image_path = 'imagem2.jpg'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

canny_image = cv2.Canny(image, 50, 200)
```

Onde é utilizado?





Filtros Lineares

- Os filtros lineares são uma classe de filtros utilizados no processamento de imagens nos quais a transformação aplicada a cada pixel é uma combinação linear dos valores dos pixels na vizinhança. Em outras palavras, a resposta do filtro em um determinado pixel é uma soma ponderada dos valores dos pixels próximos, onde os pesos são determinados pelo *kernel* do filtro.

Filtros Lineares

- Convolução
  - A convolução é uma operação matemática fundamental no processamento de imagens, usada para aplicar filtros e máscaras a uma imagem. Na convolução, um *kernel* (ou máscara) é deslizado sobre a imagem, e, em cada posição, os elementos do *kernel* são multiplicados pelos elementos correspondentes da região da imagem sob o *kernel*, e a soma desses produtos é atribuída ao pixel central. Isso é feito para todos os pixels na imagem.

Filtros Lineares

- Correlação em processamento de imagens
  - Na correlação, o *kernel* é aplicado invertendo sua forma. Em termos práticos, isso significa que, na correlação, o *kernel* se move pela imagem da esquerda para a direita, enquanto na convolução, ele se move da direita para a esquerda.

Filtros Lineares

- Exemplo de convolução

Input

0	1	2
3	4	5
6	7	8

*

Kernel

0	1
2	3

=

19	25
37	43

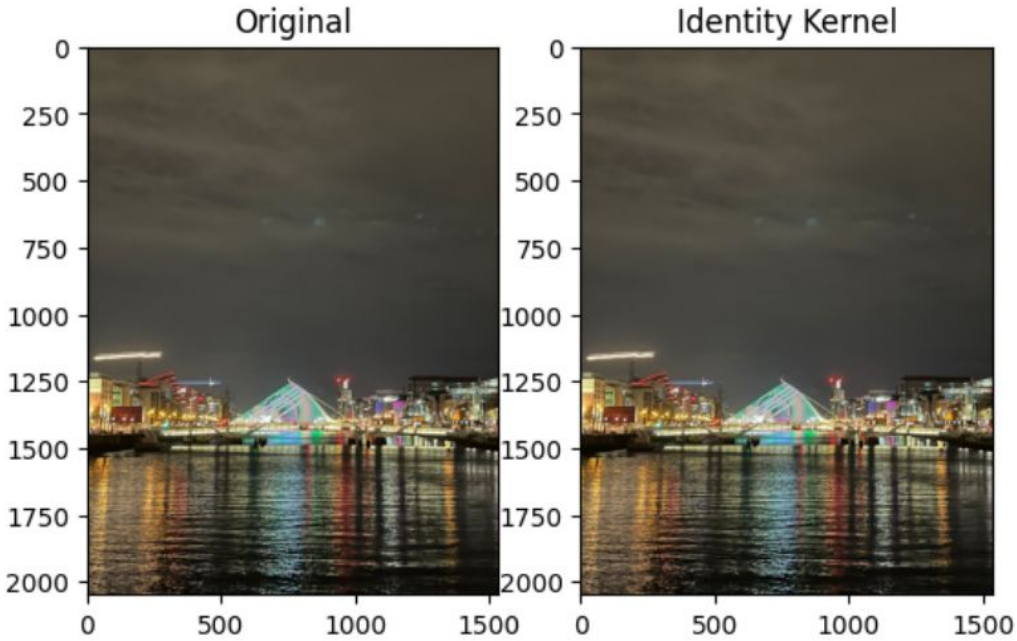
Filtros Lineares

- Exemplo de convolução - Código em python
  - Vamos aplicar o *kernel* da identidade, ou seja, não mudará nossa imagem original

```
image_path = 'imagem2.jpg'
image = cv2.imread(image_path)

kernel_identity = np.array([[0, 0, 0],
                            [0, 1, 0],
                            [0, 0, 0]])
identity = cv2.filter2D(src=image, ddepth=-1, kernel=kernel_identity)
```

depth determina o tamanho final da imagem, quando está -1, significa que será mantido o tamanho original.

Filtros Lineares



Filtros Lineares

- Os filtros, geralmente, são aplicados utilizando a lógica de convolução.
  - Por essa lógica de convolução e aplicações de filtros, foi o começo das redes neurais convolucionais, assunto que iremos ver posteriormente.
- Exemplo comuns de filtros lineares
  - Filtro de média
    - A ideia principal do filtro de média é substituir o valor de cada pixel pela média dos valores dos pixels em sua vizinhança.
  - Filtro Gaussiano
    - Ele é baseado na distribuição normal (ou gaussiana) e é especialmente eficaz na preservação de bordas enquanto suaviza outras regiões da imagem.

Filtros Lineares

- Exemplo de filtro de média

Vizinhança

5	3	6
2	9	1
8	4	7

	5	

5 + 3 + 6 + 2 + 9 + 1 + 8 + 4 + 7 = 45
45 / 9 = 5

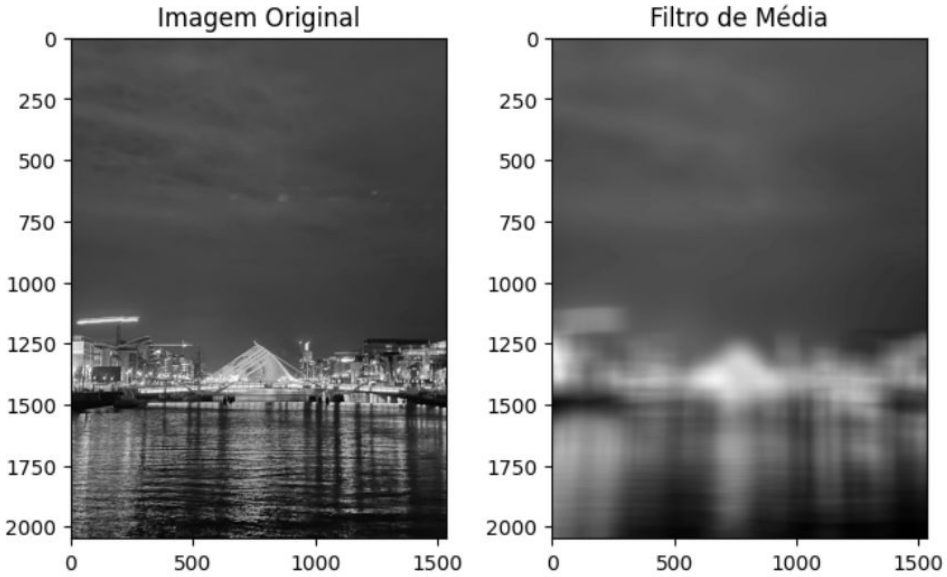
Filtros Lineares

- Exemplo de filtro de média - Python

```
image_path = "imagem2.jpg"
image = cv2.imread(image_path, cv2.IMREAD_COLOR)
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

kernel_size = 101
mean_filtered_image = cv2.blur(gray_image, (kernel_size, kernel_size))
```

Filtros Lineares

- Exemplo de filtro de média - Python



Filtros Lineares

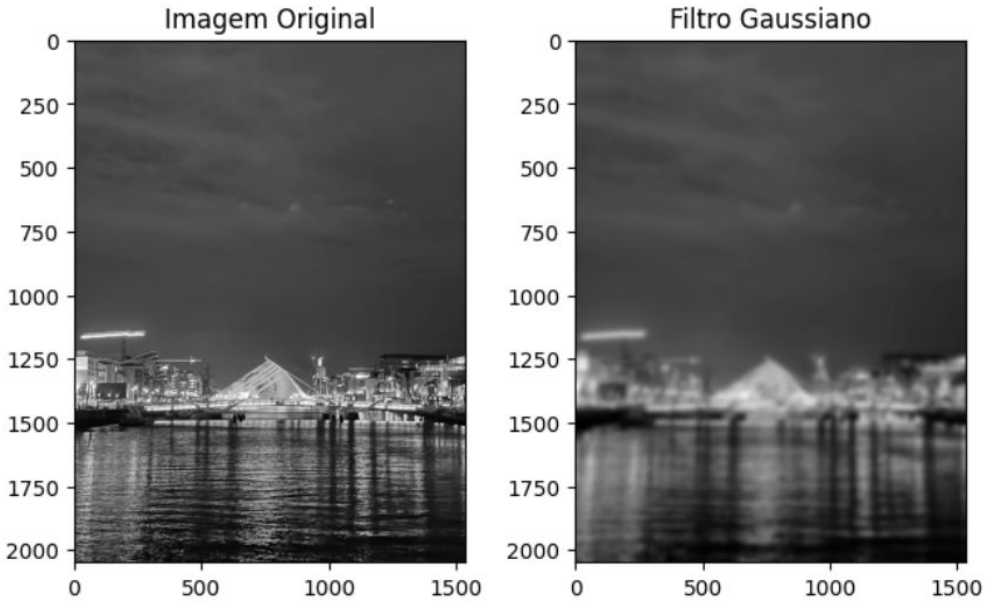
- Exemplo de filtro de Gauss- Python

```
image_path = "imagem2.jpg"
image = cv2.imread(image_path, cv2.IMREAD_COLOR)
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
sigma = 10.0
gaussian_filtered_image = cv2.GaussianBlur(gray_image, (kernel_size, kernel_size), sigma)
```

Em Gauss, utiliza-se, o desvio padrão, determinado pelo sigma e o kernel para determinar quão forte será o *blur*.

Filtros Lineares

- Exemplo de filtro de Gauss- Python



Filtros Não Lineares

- O que são?
  - Os filtros não-lineares são uma classe de filtros em processamento de imagem nos quais a saída não é uma combinação linear direta dos valores dos pixels de entrada. Em outras palavras, o resultado não é simplesmente obtido multiplicando os valores dos pixels por pesos específicos.
  - São usados, frequentemente, onde não há uniformidade nos valores dos pixels.

Filtros Não Lineares

- Filtros não lineares comuns
  - Filtro de mediana
    - Ele é usado principalmente para a remoção de ruídos do tipo impulsivo (“Salt and pepper” *noise*), preservando ao mesmo tempo as bordas da imagem. A ideia principal do filtro de mediana é substituir o valor de cada pixel pela mediana dos valores dos pixels em sua vizinhança.

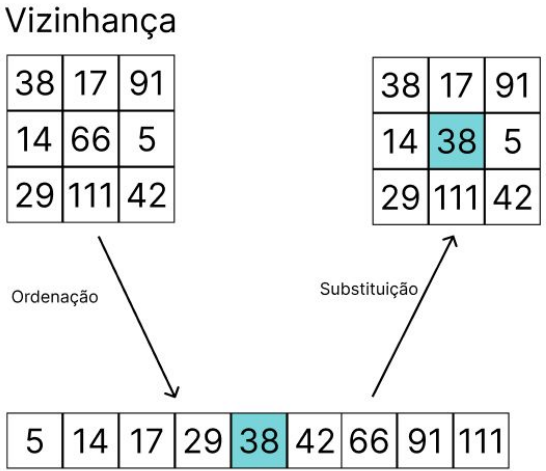
Filtros Não Lineares

- Exemplo de filtro de mediana em Python

```
image_path = "imagem2.jpg"
image = cv2.imread(image_path, cv2.IMREAD_COLOR)
kernel_size = 101
median_filtered_image = cv2.medianBlur(image, kernel_size)
```

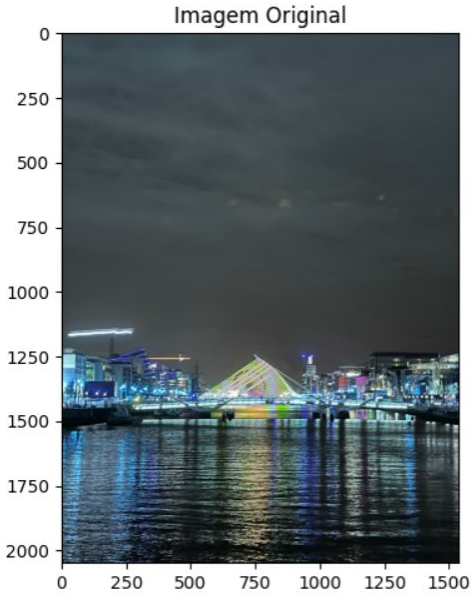
Filtros Não Lineares

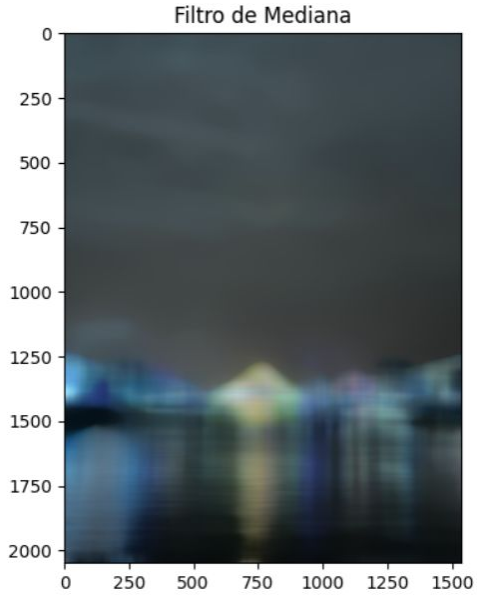
- Exemplo de filtro de mediana



Filtros Não Lineares

- Exemplo de filtro de mediana em Python





Filtros Não Lineares

- Exemplo de filtro de mediana para remoção de *salt and pepper noise*

Imagem original

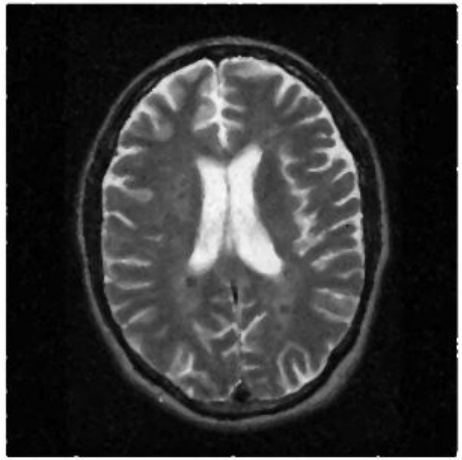


Imagem com filtro de mediana


OBRIGADO!