

ex5

October 15, 2024

1 ex5: Regularización y métodos iterativos

```
[1]: import cv2
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path
from skimage.restoration import denoise_tv_chambolle
from skimage.util import random_noise
```

```
[2]: ASSETS_FOLDER_PATH = "./assets"
OUTPUT_FOLDER_PATH = "."
```

```
[3]: Path(OUTPUT_FOLDER_PATH).mkdir(parents=True, exist_ok=True)
```

Traemos las funciones de puntos anteriores:

```
[4]: def plot_images(img1, title1, img2, title2):
    fig = plt.figure(figsize=(6, 7))

    fig.add_subplot(1, 2, 1)
    plt.imshow(img1, cmap='gray')
    plt.axis('off')
    plt.title(title1)

    fig.add_subplot(1, 2, 2)
    plt.imshow(img2, cmap='gray')
    plt.axis('off')
    plt.title(title2)

    plt.subplots_adjust(wspace=0.05, hspace=0)
    plt.show()
```

```
[5]: def mse(original, variation):
    return (original - variation).var()
```

1.1 Total Variation Regularization

```
[6]: image_path = 'assets/lena_gray.tif'  
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)  
imsize = image.shape
```

```
[7]: noise_image = random_noise(image, mode='gaussian', var=0.01)  
plot_images(image, 'Original image', noise_image, 'Noise image')
```

Original image



Noise image



```
[8]: image_denoised = denoise_tv_chambolle(noise_image, max_num_iter=10000)  
  
plot_images(noise_image, 'Noise image', image_denoised, 'Restored image')  
print(f"mse: {mse(image, image_denoised)}")
```

Noise image



Restored image



mse: 2487.330831350645