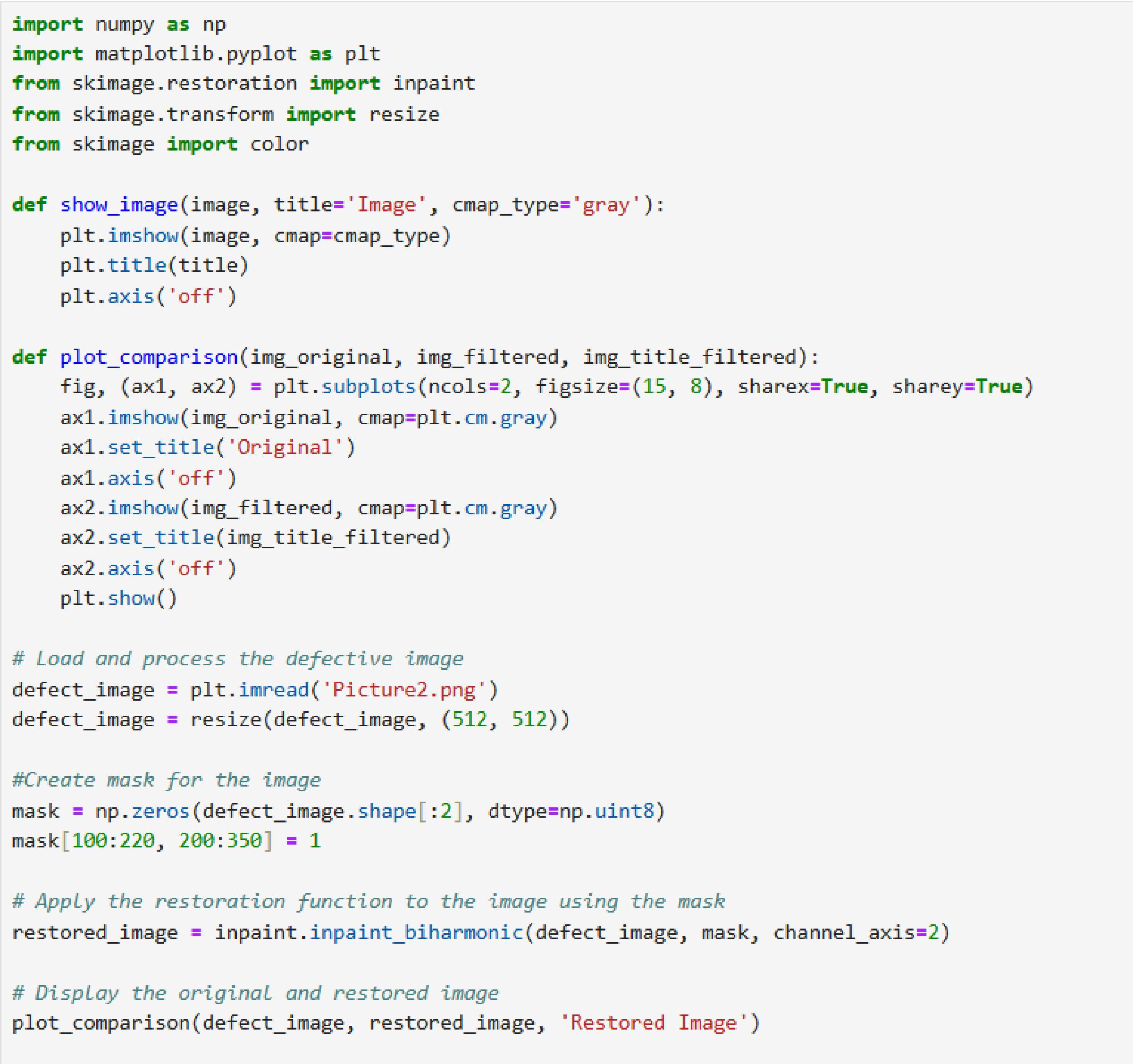
جامعة الأمير
مقرن بن عبدالعزيز
University of Prince Mugrin

AI361 – Image Processing

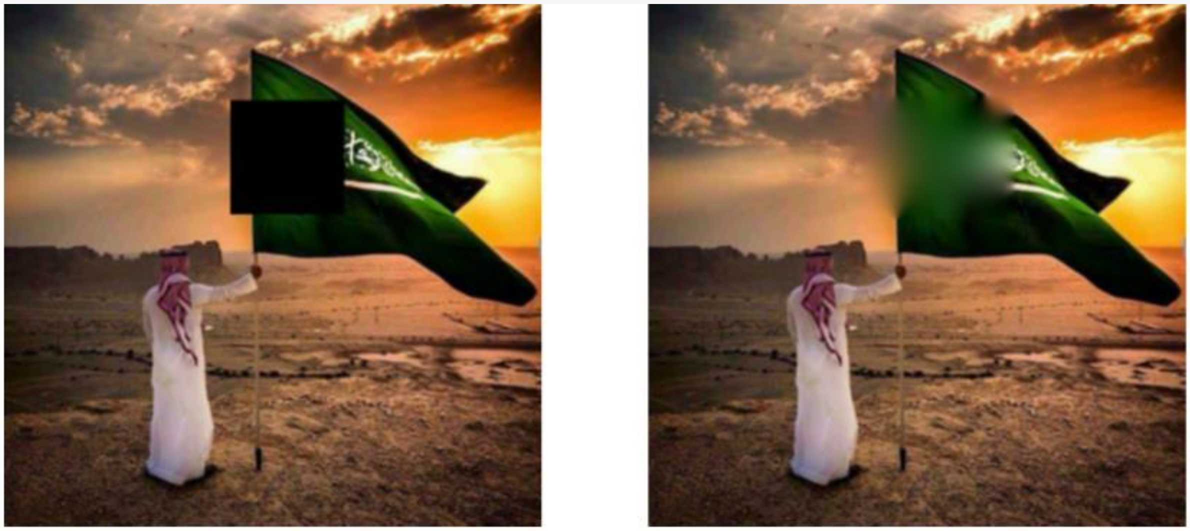
Lab 8

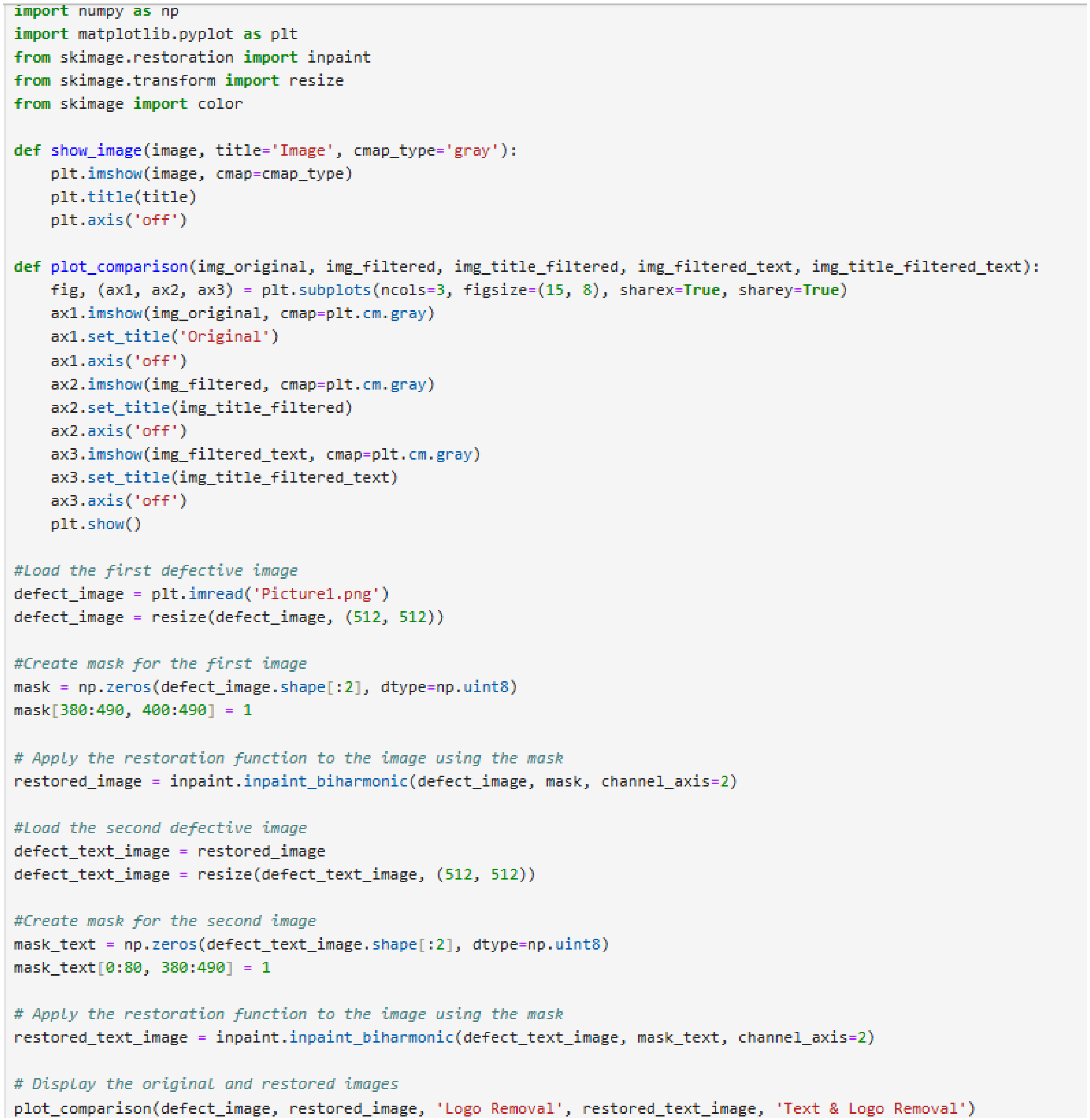
Report

Table of Contents

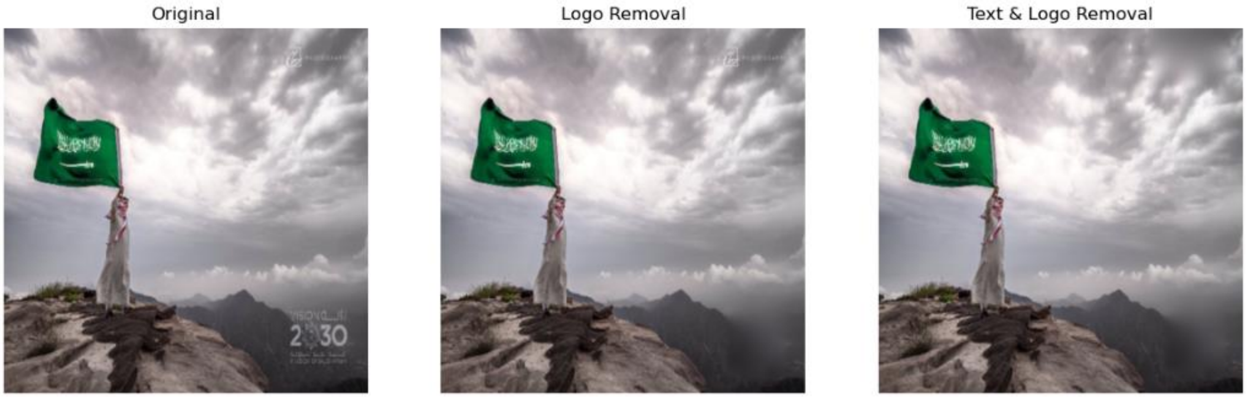


```
import numpy as np
import matplotlib.pyplot as plt
from skimage.restoration import inpaint
from skimage.transform import resize
from skimage import color

def show_image(image, title='Image', cmap_type='gray'):
    plt.imshow(image, cmap=cmap_type)
    plt.title(title)
    plt.axis('off')

def plot_comparison(img_original, img_filtered, img_title_filtered):
    fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(15, 8), sharex=True, sharey=True)
    ax1.imshow(img_original, cmap=plt.cm.gray)
    ax1.set_title('Original')
    ax1.axis('off')
    ax2.imshow(img_filtered, cmap=plt.cm.gray)
    ax2.set_title(img_title_filtered)
    ax2.axis('off')
    plt.show()

# Load and process the defective image
defect_image = plt.imread('Picture2.png')
defect_image = resize(defect_image, (512, 512))

#Create mask for the image
mask = np.zeros(defect_image.shape[:2], dtype=np.uint8)
mask[100:220, 200:350] = 1

# Apply the restoration function to the image using the mask
restored_image = inpaint.inpaint_biharmonic(defect_image, mask, channel_axis=2)

# Display the original and restored image
plot_comparison(defect_image, restored_image, 'Restored Image')
```





```
import numpy as np
import matplotlib.pyplot as plt
from skimage.restoration import inpaint
from skimage.transform import resize
from skimage import color

def show_image(image, title='Image', cmap_type='gray'):
    plt.imshow(image, cmap=cmap_type)
    plt.title(title)
    plt.axis('off')

def plot_comparison(img_original, img_filtered, img_title_filtered, img_filtered_text, img_title_filtered_text):
    fig, (ax1, ax2, ax3) = plt.subplots(ncols=3, figsize=(15, 8), sharex=True, sharey=True)
    ax1.imshow(img_original, cmap=plt.cm.gray)
    ax1.set_title('Original')
    ax1.axis('off')
    ax2.imshow(img_filtered, cmap=plt.cm.gray)
    ax2.set_title(img_title_filtered)
    ax2.axis('off')
    ax3.imshow(img_filtered_text, cmap=plt.cm.gray)
    ax3.set_title(img_title_filtered_text)
    ax3.axis('off')
    plt.show()

#Load the first defective image
defect_image = plt.imread('Picture1.png')
defect_image = resize(defect_image, (512, 512))

#Create mask for the first image
mask = np.zeros(defect_image.shape[:2], dtype=np.uint8)
mask[380:490, 400:490] = 1

# Apply the restoration function to the image using the mask
restored_image = inpaint.inpaint_biharmonic(defect_image, mask, channel_axis=2)

#Load the second defective image
defect_text_image = restored_image
defect_text_image = resize(defect_text_image, (512, 512))

#Create mask for the second image
mask_text = np.zeros(defect_text_image.shape[:2], dtype=np.uint8)
mask_text[0:80, 380:490] = 1

# Apply the restoration function to the image using the mask
restored_text_image = inpaint.inpaint_biharmonic(defect_text_image, mask_text, channel_axis=2)

# Display the original and restored images
plot_comparison(defect_image, restored_image, 'Logo Removal', restored_text_image, 'Text & Logo Removal')
```



Original

Logo Removal

Text & Logo Removal

Why did I put the mask = 1?

Because it is a binary image and when I put the mask = 1, then I’m telling the inpaint function that those coordinates who are set to 1 is the one we need to change, and everything else is 0 so the inpaint function will not do anything to it

What is the inpaint function?

The inpaint function in the skimage library helps to restore areas of an image. It takes the original image and the mask, then tries to fill in the masked regions by blending in surrounding pixels.

Summary

I used the same functions and the same code for both images (from the Lab08-Manual).

But in the second one I added another mask so that I can double the adjusting on the image because it has Text & Logo, while the first one has only black box in the middle, and I wanted to remove them all.

I created a mask = 1 with the black box ,logo, and text coordinates and applied it using inpaint function from skimage library