Homework 2 - TV reconstruction

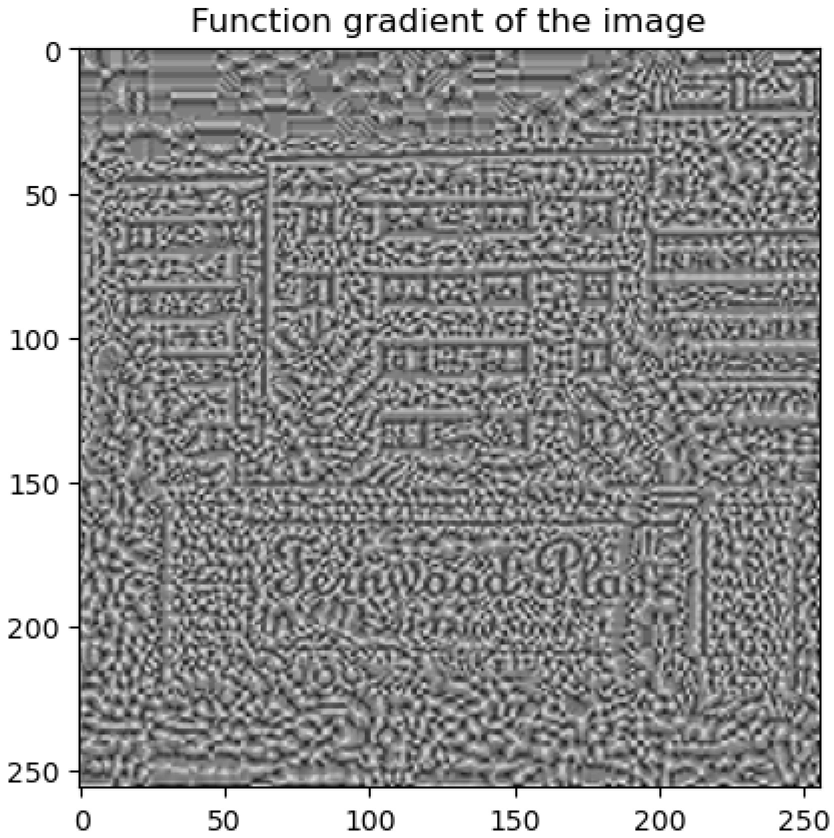
In [1]:

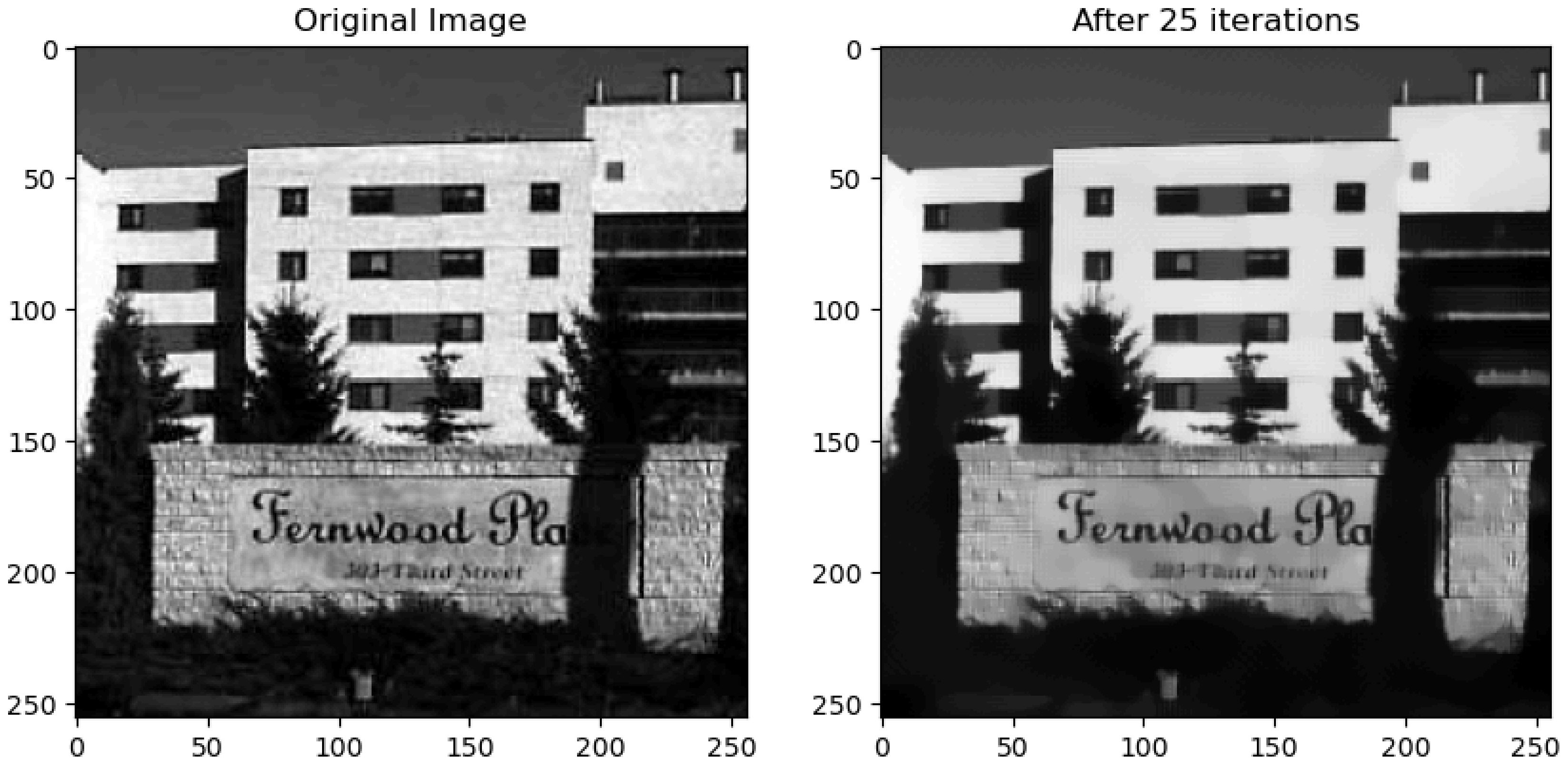
```
import numpy as np
import matplotlib.pyplot as plt
import autograd.numpy as anp
from autograd import grad
from skimage import io, color
```

In [2]:

```
# Imports the image
img = io.imread("image_0303.jpg")

# Convert the image into grayscale
if img.ndim == 3:
    img = color.rgb2gray(img)

# Define a function for calculating Total Variation of the image
def Total_Variation(image):
    gx, gy = anp.gradient(image)
    gradient = anp.sqrt(anp.abs(gx)**2 + anp.abs(gy)**2 + 1e-10)
    return anp.sum(gradient)

TV = Total_Variation(img)
print(f"Numerical Value of Total variation of the image : {TV}")

# Compute functional gradient of the image
func_gradient = grad(Total_Variation, 0)
h = func_gradient(img)
# Display the functional gradient
plt.figure()
plt.imshow(h, cmap = "gray")
plt.title("Function gradient of the image")
plt.show()

# TV reconstruction loop
g = img # Input image
for i in range(25):
    h = func_gradient(g) # Computes function gradient for the input image
    gnew = g - 5*1e-3*anp.linalg.norm(g)*h/anp.linalg.norm(h) # TV reconstruction
    g = gnew # Update the image

# Display the results
plt.figure(figsize = (10,5))
plt.subplot(1,2,1)
plt.imshow(img, cmap = "gray")
plt.title("Original Image")
plt.subplot(1,2,2)
plt.imshow(gnew, cmap = "gray")
plt.title(f"After 25 iterations")
plt.show()
```

```
Numerical Value of Total variation of the image : 1094864.8904494494
```





Original Image

After 25 iterations

We note that after 25 iterations using TV reconstruction, the parts of the image are smoothened where the gradient is small (e.g. The texture on the building wall and the brick wall around the text "Fernwood Plaza"). The edges, where the gradient is large, are preserved.