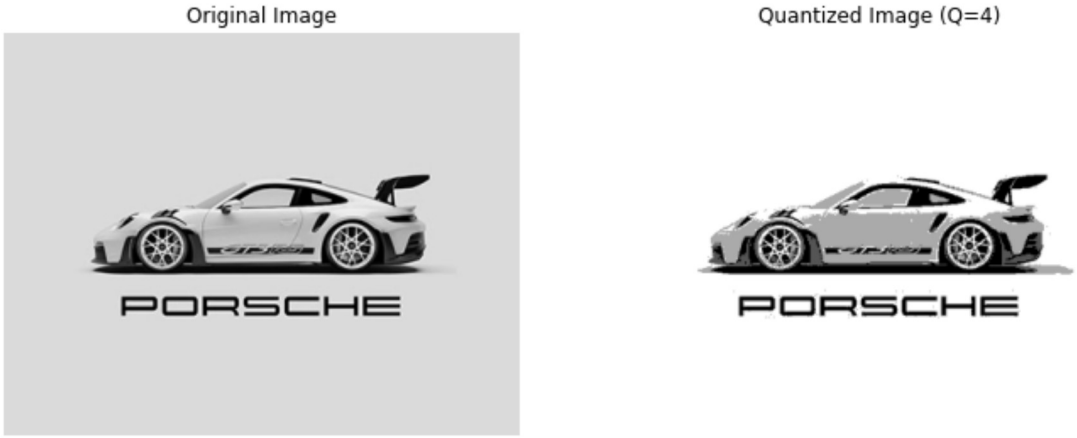
In [1]:
```
# Experiment 3.1 : Image sampling and quantization

import numpy as np
import matplotlib.pyplot as plt
from skimage import io, color

# Step 1: Load the grayscale image I
image_path = 'C:/Users/Student/Documents/Exp1image.jpg'
I = io.imread(image_path)

# Convert to grayscale if the image is not already
I_gray = color.rgb2gray(I) if I.ndim == 3 else I

# Step 2: Define the sampling factor F and initialize the sampled image Is
F = 2  # Define the sampling factor (you can change this value)

# Get the dimensions of the original image
height, width = I_gray.shape

# Create a new image that will hold the sampled data
Is = I_gray[::F, ::F]

# Step 3: Plot the original and sampled images side by side
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Show the original image
axes[0].imshow(I_gray, cmap='gray')
axes[0].set_title('Original Image')
axes[0].axis('off')

# Show the sampled image
axes[1].imshow(Is, cmap='gray')
axes[1].set_title(f'Sampled Image (F={F})')
axes[1].axis('off')

# Show the plots
plt.show()
```

In [2]:
```
# Experiment 3.2 : Image sampling and quantization

import numpy as np
import matplotlib.pyplot as plt
from skimage import io, color

# Step 1: Load the grayscale image I
image_path = 'C:/Users/Student/Documents/Exp1image.jpg'
I = io.imread(image_path)

# Convert to grayscale if the image is not already
I_gray = color.rgb2gray(I) if I.ndim == 3 else I

# Step 2: Define the number of quantization levels Q
Q = 4

# Step 3: Normalize pixel values to the range [0, 1]
I_prime = I_gray  # I_gray already has values in the range [0, 1], no need to di

# Step 4: Scale the normalized values to the range [0, Q-1] and round
Iq_prime = np.round(I_prime * (Q - 1))

# Step 5: Scale back to [0, 255]
Iq = np.uint8(Iq_prime * (255 / (Q - 1)))  # Ensure values are scaled to [0, 255

# Step 6: Plot the original and quantized images side by side for comparison
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Show the original image
axes[0].imshow(I_gray, cmap='gray')
axes[0].set_title('Original Image')
axes[0].axis('off')

# Show the quantized image
axes[1].imshow(Iq, cmap='gray')
axes[1].set_title(f'Quantized Image (Q={Q})')
axes[1].axis('off')

# Show the plots
plt.show()
```



In [ ]: