```python
#11
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Load an image and convert it to grayscale
image = cv2.imread('/content/DALL·E 2024-11-14 14.41.04 - Create a
high-quality image of a colorful outdoor scene with diverse elements.
Include a small house or cabin with a wooden texture, surrounded by
gre.webp', cv2.IMREAD_GRAYSCALE)
if image is None:
    raise FileNotFoundError("The specified image path is incorrect or
the image does not exist.")

# Define Gabor filter parameters
ksize = 31          # Size of the filter
sigma = 4.0         # Standard deviation of the Gaussian function
lambd = 10.0        # Wavelength of the sinusoidal factor
gamma = 0.5         # Spatial aspect ratio
psi = 0             # Phase offset

# Create multiple Gabor filters with varying orientations
angles = [0, 45, 90, 135]  # Orientations in degrees
gabor_outputs = []

for theta in angles:
    theta_rad = np.deg2rad(theta)
    kernel = cv2.getGaborKernel((ksize, ksize), sigma, theta_rad,
lambd, gamma, psi, ktype=cv2.CV_32F)

    # Apply Gabor filter to the image
    filtered_image = cv2.filter2D(image, cv2.CV_8UC1, kernel)  #
Change to CV_8UC1 for grayscale
    gabor_outputs.append((theta, filtered_image))

# Display the original and filtered images
plt.figure(figsize=(12, 8))
plt.subplot(2, 3, 1)
plt.imshow(image, cmap='gray')
plt.title("Original Image")
plt.axis("off")

# Plot the results for each orientation
for i, (theta, filtered_image) in enumerate(gabor_outputs, start=2):
    plt.subplot(2, 3, i)
    plt.imshow(filtered_image, cmap='gray')
    plt.title(f"Gabor Filter - {theta}°")
    plt.axis("off")
```

```
plt.tight_layout()
plt.show()
```