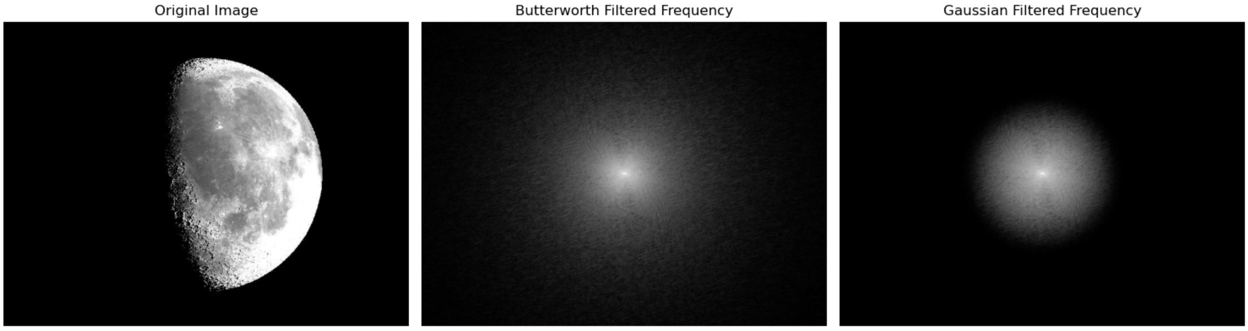
In [3]:

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from scipy.ndimage import uniform_filter
def fourier_transform(image):
    return np.fft.fftshift(np.fft.fft2(image))


def butterworth_filter(shape, cutoff, order=2):
    rows, cols = shape
    x = np.arange(-cols // 2, cols // 2)
    y = np.arange(-rows // 2, rows // 2)
    X, Y = np.meshgrid(x, y)
    D = np.sqrt(X**2 + Y**2)
    H = 1 / (1 + (D / cutoff)**(2 * order))
    return H


def gaussian_filter(shape, cutoff):
    rows, cols = shape
    x = np.arange(-cols // 2, cols // 2)
    y = np.arange(-rows // 2, rows // 2)
    X, Y = np.meshgrid(x, y)
    D = np.sqrt(X**2 + Y**2)
    H = np.exp(-(D**2) / (2 * (cutoff ** 2)))
    return H


image_path = '/Users/karedlashilpa/Downloads/Moon_Image.jpg'
img = Image.open(image_path).convert('L')
image_np = np.array(img)
f_transform = fourier_transform(image_np)

#I am Creating Butterworth and Gaussian filters here
butterworth = butterworth_filter(image_np.shape, cutoff=30, order=2)
gaussian = gaussian_filter(image_np.shape, cutoff=30)
filtered_butterworth = f_transform * butterworth
filtered_gaussian = f_transform * gaussian
fig, axes = plt.subplots(1, 3, figsize=(15, 5))


axes[0].imshow(image_np, cmap='gray')
axes[0].set_title('Original Image')
axes[0].axis('off')
axes[1].imshow(np.log(np.abs(filtered_butterworth) + 1), cmap='gray')
axes[1].set_title('Butterworth Filtered Frequency')
axes[1].axis('off')
axes[2].imshow(np.log(np.abs(filtered_gaussian) + 1), cmap='gray')
axes[2].set_title('Gaussian Filtered Frequency')
axes[2].axis('off')
plt.tight_layout()
plt.show()
```



Original Image

Butterworth Filtered Frequency

Gaussian Filtered Frequency

In [ ]: