Blur the Input Image

In [1]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
img = cv2.imread(r'C:\Users\91944\Downloads\image.jpg')
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
f = np.fft.fft2(gray_img)
fshift = np.fft.fftshift(f)
rows, cols = gray_img.shape
crow, ccol = rows // 2, cols // 2
x, y = np.ogrid[-crow:rows - crow, -ccol:cols - ccol]
gaussian_filter = np.exp(-(x ** 2 + y ** 2) / (2 * 25 ** 2))
filtered_spectrum = fshift * gaussian_filter
filtered_img = np.abs(np.fft.ifft2(np.fft.ifftshift(filtered_spectrum)))
filtered_gray_img = np.uint8(filtered_img)
fig, axs = plt.subplots(1, 2)
axs[0].imshow(gray_img, cmap='gray')
axs[0].set_title('Input Image')
axs[1].imshow(filtered_gray_img, cmap='gray')
axs[1].set_title('Blurred Image')
plt.show()
```

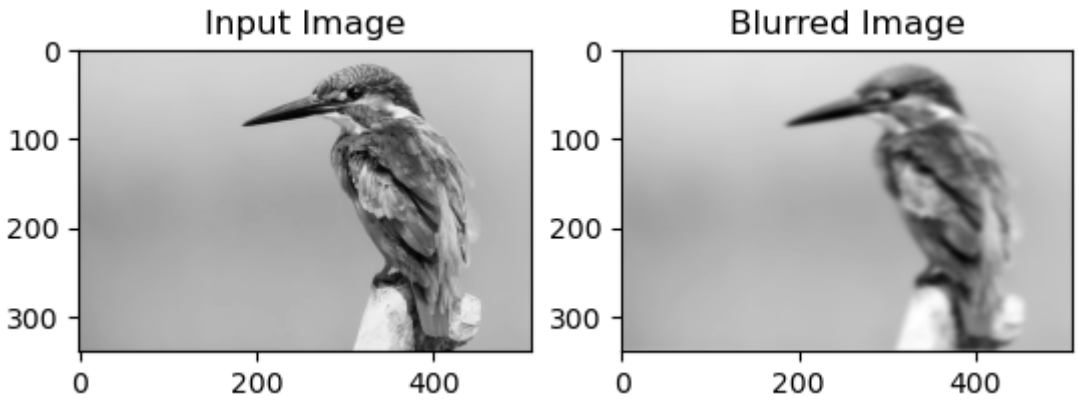

Image Restoration

In [2]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
img = cv2.imread(r'C:\Users\91944\Downloads\download.jpeg', cv2.IMREAD_GRAYSCALE)
f = np.fft.fft2(img)
fshift = np.fft.fftshift(f)
magnitude_spectrum = 20 * np.log(np.abs(fshift))
rows, cols = img.shape
crow, ccol = rows // 2, cols // 2
r = 30
mask = np.zeros((rows, cols), np.uint8)
cv2.circle(mask, (ccol, crow), r, 1, -1)
fshift_filtered = fshift * mask
fshift_filtered_back = np.fft.ifftshift(fshift_filtered)
img_filtered = np.fft.ifft2(fshift_filtered_back)
img_filtered = np.real(img_filtered)
img_filtered = np.uint8(np.clip(img_filtered, 0, 255))
plt.subplot(121), plt.imshow(img, cmap='gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(img_filtered, cmap='gray')
plt.title('Filtered Image'), plt.xticks([]), plt.yticks([])
plt.show()
```

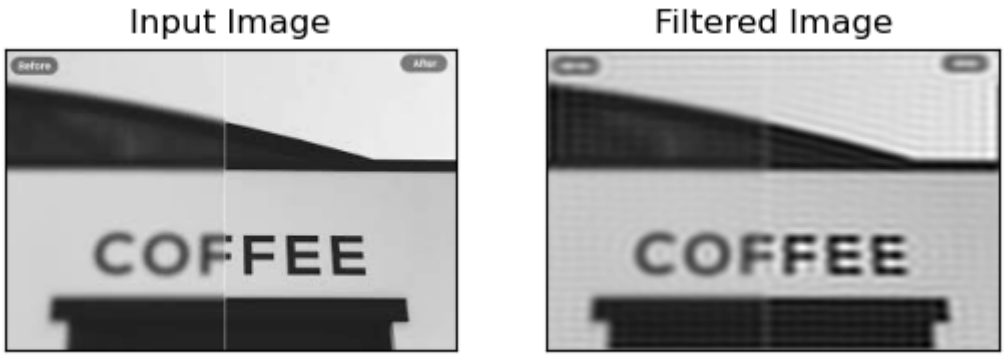

Image Compression

In [2]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
img = cv2.imread(r'C:\Users\91944\Downloads\image.jpg')
print('Size of original image:', img.size,'bytes')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
f = np.fft.fft2(gray)
T = 0.1 * np.amax(np.abs(f))
f_filtered = np.multiply(np.abs(f) > T, f)
img_compressed = np.real(np.fft.ifft2(f_filtered))
print('Size of compressed image:', img_compressed.size,'bytes')
```

```
Size of original image: 520200 bytes
Size of compressed image: 173400 bytes
```

In [ ]: