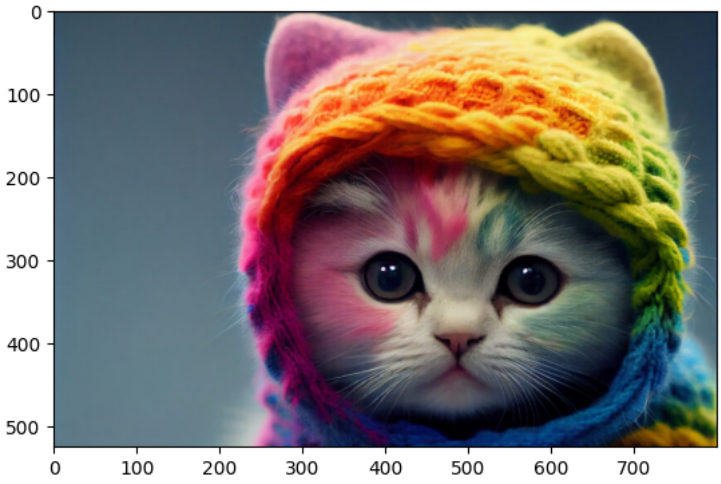
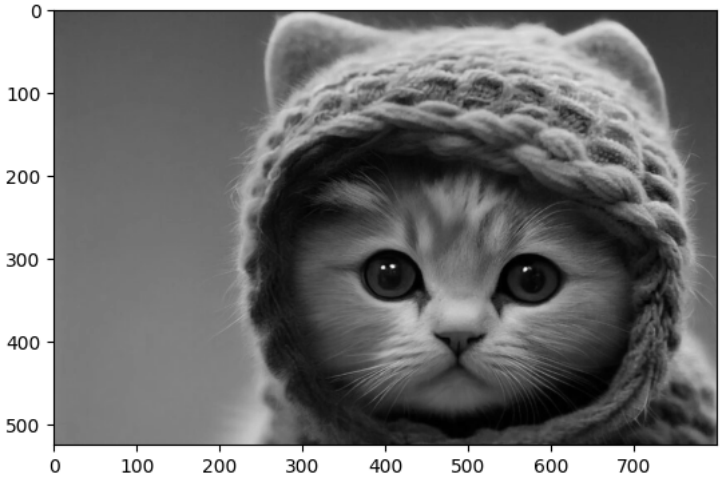
```
import cv2
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
import numpy as np
import math
import random


img = cv2.imread('/content/bb.jpg')


img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7f1c4d2ee580>
```



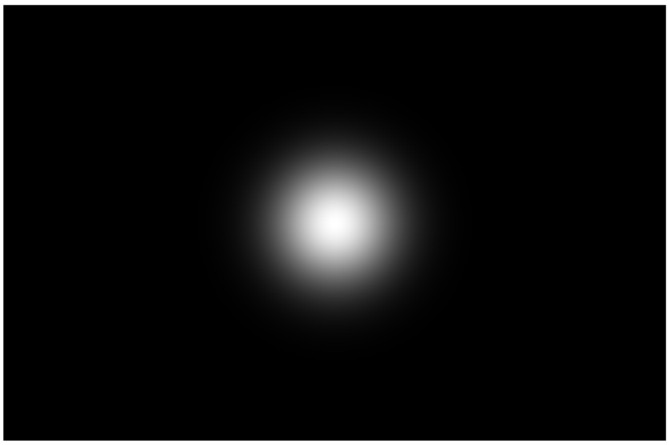

```
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(img , cmap = "gray")
```

```
<matplotlib.image.AxesImage at 0x7f1c4c26f460>
```




```
F = np.fft.fft2(img)
Fshift = np.fft.fftshift(F)
M,N = img.shape
H = np.zeros((M,N), dtype=np.float32)


D0 = 50
for u in range(M):
    for v in range(N):
        D = np.sqrt((u-M/2)**2 + (v-N/2)**2)
        H[u,v] = np.exp(-(D**2)/(2*(D0**2)))

plt.imshow(H, cmap='gray')
plt.axis('off')
plt.show()

Gshift = Fshift * H
G = np.fft.ifftshift(Gshift)
g = np.abs(np.fft.ifft2(G))
plt.imshow(g, cmap='gray')
plt.axis('off')
```

```
plt.axis('off')
plt.show()
```



```
H = 1 - H
plt.imshow(H, cmap='gray')
plt.axis('off')
plt.show()
Gshift = Fshift * H
G = np.fft.ifftshift(Gshift)
g = np.abs(np.fft.ifft2(G))
plt.imshow(g, cmap='gray')
plt.axis('off')
plt.show()
```

Conclusion:

Gaussian Lowpass Filter is used for image smoothing in the frequency domain. As seen clearly from the output, it removes high-frequency noise from a digital image and preserves low-frequency components. But here it eliminates ringing effect while blurring down the image.

Gaussian Highpass Filter (IHPF) is used for image sharpening in the frequency domain. As seen clearly from the output, it removes low-frequency components from an image and preserves high-frequency components.