EXPT NO: 5	**FOURIER TRANSFORM FOR FILTERING THE IMAGE**
DATE: 01/08/2025	

AIM:

To implement the Use of Fourier transform for filtering the image.

ALGORITHM:

1. Convert image to **grayscale** and **float type.**
2. Apply DFT using **np.fft.fft2()** and shift using **np.fft.fftshift().**
3. Create a mask for low/high pass filtering in frequency domain.
4. Apply mask and perform inverse FFT using **np.fft.ifft2().**
5. Take **magnitude** and **display filtered image**.
6. Compare with **original.**

CODE:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Read image
img = cv2.imread('/content/drive/MyDrive/input.jpg', 0)

# Check if the image was loaded successfully
if img is None:
    print("Error: Could not load image. Please check the file path and ensure
the image exists.")
else:
    # Fourier Transform
    f = np.fft.fft2(img)
    fshift  = np.fft.fftshift(f)
    magnitude_spectrum = 20*np.log(np.abs(fshift))
```

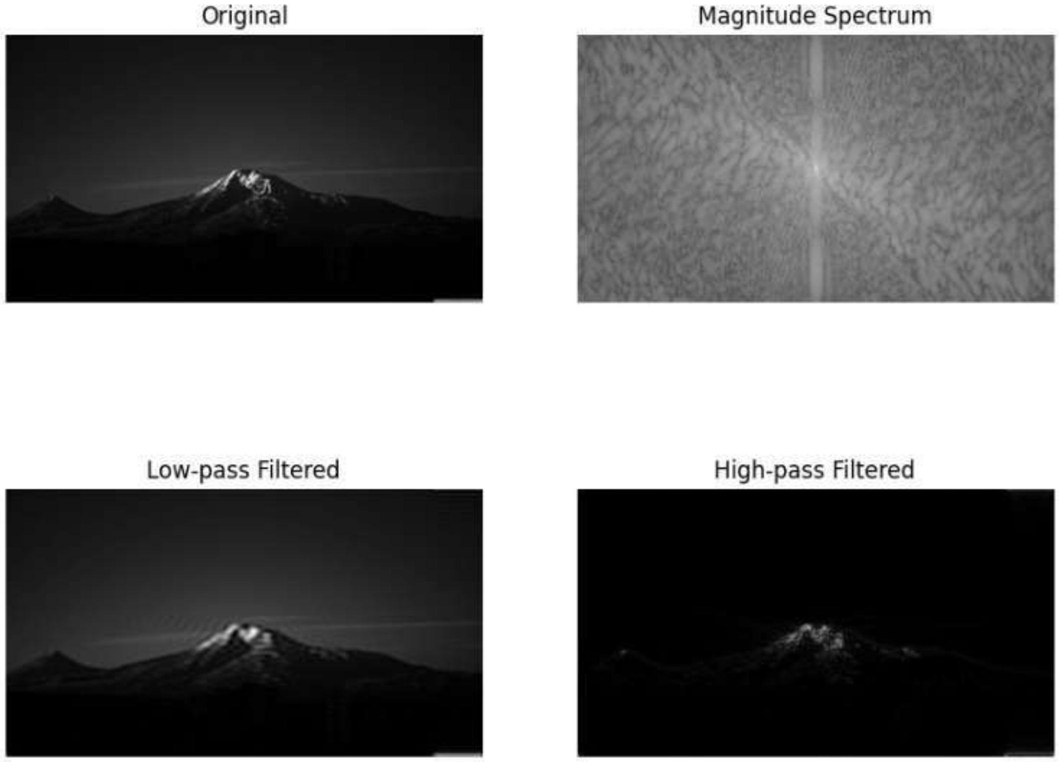
```
# --- Low-pass filter ---
rows, cols = img.shape
crow, ccol = rows//2, cols//2
mask_lp = np.zeros((rows, cols), np.uint8)
r = 50  # radius for LPF
cv2.circle(mask_lp, (ccol, crow), r, 1, -1)

fshift_lp = fshift * mask_lp
img_lp = np.fft.ifft2(np.fft.ifftshift(fshift_lp))
img_lp = np.abs(img_lp)

# --- High-pass filter ---
mask_hp = 1 - mask_lp
fshift_hp = fshift * mask_hp
img_hp = np.fft.ifft2(np.fft.ifftshift(fshift_hp))
img_hp = np.abs(img_hp)

# Display results
plt.figure(figsize=(10,8))
plt.subplot(2,2,1), plt.imshow(img, cmap='gray')
plt.title("Original"), plt.axis('off')
plt.subplot(2,2,2), plt.imshow(magnitude_spectrum, cmap='gray')
plt.title("Magnitude Spectrum"), plt.axis('off')
plt.subplot(2,2,3), plt.imshow(img_lp, cmap='gray')
plt.title("Low-pass Filtered"), plt.axis('off')
plt.subplot(2,2,4), plt.imshow(img_hp, cmap='gray')
plt.title("High-pass Filtered"), plt.axis('off')
plt.show()
```

OUTPUT:



RESULT:

Thus, Fourier transform for filtering the image was implemented successfully.