EXPT 4	Advanced Filters

AIM:-

To implement advanced filtering techniques including Laplacian filter for edge detection, Difference of Gaussian (DoG) for feature extraction, and Laplacian of Gaussian (LoG) for blob detection and edge enhancement.

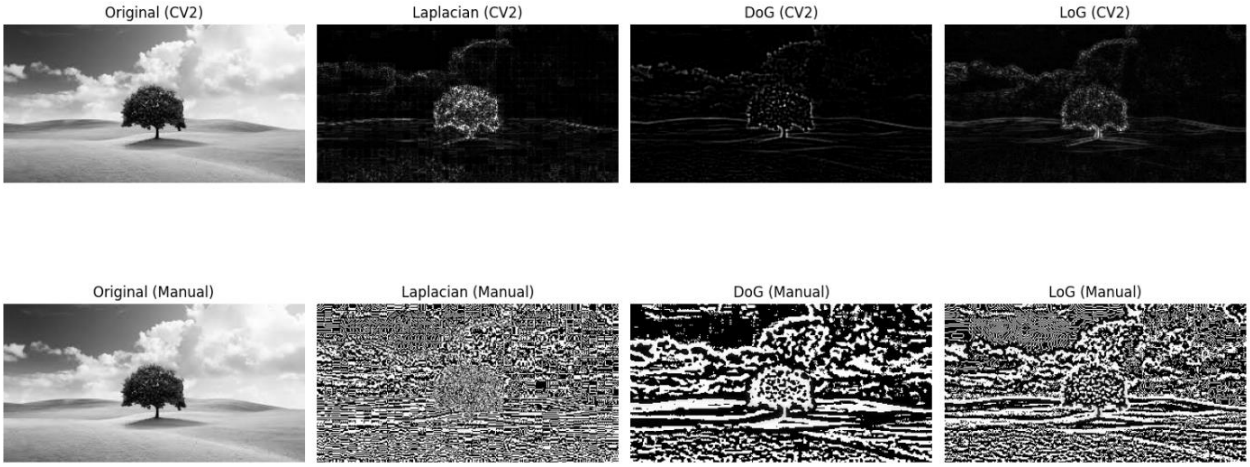
CODE:-

```
import cv2
from google.colab.patches import cv2_imshow
import numpy as np
from scipy import ndimage
import matplotlib.pyplot as plt
from skimage import io

img_cv2 = cv2.imread('input.jpg', cv2.IMREAD_GRAYSCALE)
img_skimage = io.imread('input.jpg', as_gray=True)
img_skimage = (img_skimage * 255).astype(np.uint8)

if img_cv2 is None or img_skimage is None:
    print("Upload 'input.jpg' to Colab.")
else:
    # Laplacian Filter (CV2)
    laplacian_cv2 = cv2.Laplacian(img_cv2, cv2.CV_64F)
    laplacian_abs_cv2 = cv2.convertScaleAbs(laplacian_cv2)

    # Laplacian Filter (Manual)
    laplacian_kernel = np.array([[0,1,0],[1,-4,1],[0,1,0]])
    laplacian_manual = ndimage.convolve(img_skimage, laplacian_kernel)
    laplacian_abs_manual = np.absolute(laplacian_manual).astype(np.uint8)

    # Difference of Gaussian (DOG) (CV2)
    blur1_cv2 = cv2.GaussianBlur(img_cv2, (5,5), 1)
```

```
    blur2_cv2 = cv2.GaussianBlur(img_cv2, (5,5), 2)
    dog_cv2 = cv2.subtract(blur1_cv2, blur2_cv2)

    # Difference of Gaussian (DOG) (Manual)
    blur1_manual = ndimage.gaussian_filter(img_skimage, sigma=1)
    blur2_manual = ndimage.gaussian_filter(img_skimage, sigma=2)
    dog_manual = blur1_manual - blur2_manual
    dog_norm_manual = ((dog_manual - dog_manual.min()) / (dog_manual.max() -
dog_manual.min()) * 255).astype(np.uint8)

    # Laplacian of Gaussians (LOG) (CV2)
    blur_cv2 = cv2.GaussianBlur(img_cv2, (5,5), 1)
    log_cv2 = cv2.Laplacian(blur_cv2, cv2.CV_64F)
    log_abs_cv2 = cv2.convertScaleAbs(log_cv2)

    # Laplacian of Gaussians (LOG) (Manual)
    blur_manual = ndimage.gaussian_filter(img_skimage, sigma=1)
    log_manual = ndimage.laplace(blur_manual)
    log_abs_manual = np.absolute(log_manual)
    log_norm_manual = (log_abs_manual / log_abs_manual.max() *
255).astype(np.uint8)

    # Display images
    plt.figure(figsize=(15, 8))

    plt.subplot(2, 4, 1)
    plt.title('Original (CV2)')
    plt.imshow(img_cv2, cmap='gray')
    plt.axis('off')

    plt.subplot(2, 4, 2)
    plt.title('Laplacian (CV2)')
    plt.imshow(laplacian_abs_cv2, cmap='gray')
    plt.axis('off')
```

```
plt.subplot(2, 4, 3)
plt.title('DoG (CV2)')
plt.imshow(dog_cv2, cmap='gray')
plt.axis('off')

plt.subplot(2, 4, 4)
plt.title('LoG (CV2)')
plt.imshow(log_abs_cv2, cmap='gray')
plt.axis('off')

plt.subplot(2, 4, 5)
plt.title('Original (Manual)')
plt.imshow(img_skimage, cmap='gray')
plt.axis('off')

plt.subplot(2, 4, 6)
plt.title('Laplacian (Manual)')
plt.imshow(laplacian_abs_manual, cmap='gray')
plt.axis('off')

plt.subplot(2, 4, 7)
plt.title('DoG (Manual)')
plt.imshow(dog_norm_manual, cmap='gray')
plt.axis('off')

plt.subplot(2, 4, 8)
plt.title('LoG (Manual)')
plt.imshow(log_norm_manual, cmap='gray')
plt.axis('off')

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

OUTPUT:-



RESULT:-

Successfully applied advanced filters demonstrating their effectiveness in detecting edges, blobs, and image features. Each filter showed distinct characteristics suitable for specific computer vision preprocessing and feature extraction tasks.