Ex: No: 9

Date:

Fuzzy logic

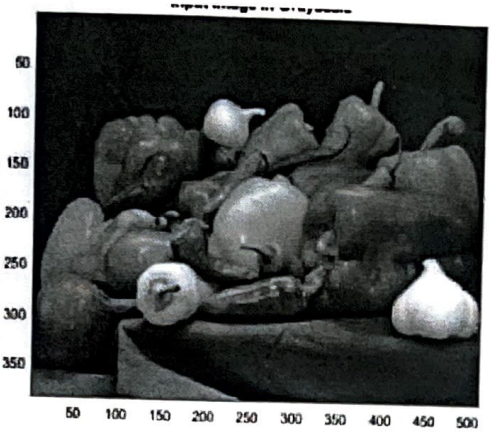
Program code:

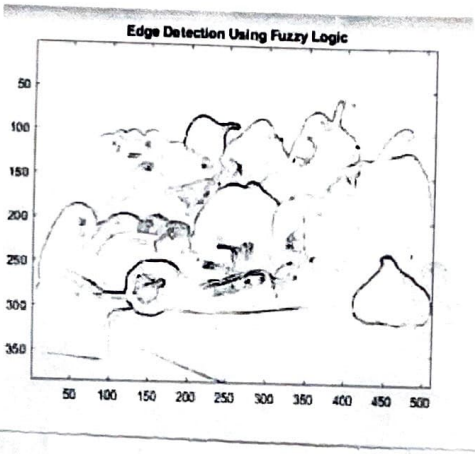
```
Import numpy as np
import matplotlib.pyplot as plt
from skimage import io, color.
from scipy.ndimage import convolve.
import skfuzzy as fuzz.

Irgb = io.imread(" ")
I gray = color.rgb2gray(I rgb)

plt.figure()
plt.imshow(I gray, cmap='gray')
plt.title('Input Image in Grayscale')
plt.axis('off')
plt.show().

I = Igray.astype(np.float64)
```

```
Gx = np.array([[-1, 1]])
Gy = Gx.T

Ix = convolve(I, Gx, mode='nearest')

Iy = convolve(I, Gy, mode='nearest')
plt.figure()

plt.imshow(Ix, cmap='gray')

plt.title('Ix (Gradient in x direction)')
plt.axis('off')

plt.show()

x = np.linspace(-1, 1, 100)

Ix_zero = fuzz.gaussmf(x, mean=0, sigma=0.1)

Iy_zero = fuzz.gaussmf(x, mean=0, sigma=
0.1)

for i in range(I.shape[0]):

    for j in range(I.shape[1]):

        ix_val = Ix[i, j]
        iy_val = Iy[i, j]

Ix_mem = fuzz.interp_membership(x, Ix_zero,
ix_val)
```

```
Iy_mem = fuzz.interp_membership(x, Iy_zero, iy_val)

if Ix_mem > 0 and Iy_mem > 0:
    Iout_mem = fuzz.interp_membership(x, Iout_white, 1)
else:
    Iout_mem = fuzz.interp_membership(x, Iout_black, 1)

Ieval[i, j] = Iout_mem

plt.figure()
plt.imshow(I, cmap='gray')
plt.title('original Grayscale Image')
plt.axis('off')
plt.show()

plt.figure()
plt.imshow(I, cmap='gray')
plt.title('Edge detection Using fuzzy logic')
plt.axis('off')
plt.show()
```

output :





Result :

Thus the program is successfully executed & the output is verified.