**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Department of Electronics and Telecommunication Engineering**

**Subject: Image and Video Processing Program: B.Tech/BTI/MBA**

**Sem: VII/IX/V ACAY: 2020-21**

**EXPERIMENT NO. 8**

**Aim:**

To write a program in PYTHON to highlight horizontal, vertical and diagonal edges of an image

**Software:**  PYTHON.

**Prerequisite:**

|  |  |
| --- | --- |
| Sr. No | Concepts |
| 1. | Sobel and Laplacian operators |

**Outcome:**

After successful completion of this experiment students will be able to:

1. Understand the significance of filter masks for edge enhancement
2. Implement Sobel and Laplacian operators

**Theory:**

**Sobel Operator**

|  |  |  |
| --- | --- | --- |
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

**Horizontal edge Fx**

|  |  |  |
| --- | --- | --- |
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

**Vertical Edge Fy**

* Convolve Fx mask to the original image to obtain the x gradient of the image
* Convolve Fy mask to the original image to obtain the y gradient of the image
* Add the results of the above two steps

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| -1 | 0 | 1 |
| -2 | -1 | 0 |

**Diagonal Edge**

**Laplacian operator**

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

|  |
| --- |
| Name of the Experiment: To implement Sobel operators on an image to detect edges |
| Roll No. C018 Name: Dhruvit Jain |
| Program: B.Tech ExTC Semester : VII |
| Date of Performance:04/09/2020 Date of Submission: 04/09/2020 |

**CODE:**

import numpy as np

from skimage import io

import matplotlib.pyplot as plt

from scipy import signal

from skimage.color import rgb2gray

image\_ori=io.imread('boy.bmp')

image\_ori = rgb2gray(image\_ori)

io.imshow (image\_ori)

sh= image\_ori.shape



horz =[[1,2,1],[0,0,0],[-1,-2,-1]]

horz\_img = signal.convolve2d(image\_ori,horz,mode='same')

for r in range (0,sh[0]):

for c in range (0,sh[1]):

if horz\_img [r][c] < 0:

horz\_img [r][c]= 0

io.imshow (horz\_img, cmap='gray')



vert =[[-1,0,1],[-2,0,2],[-1,0,1]]

vert\_img = signal.convolve2d(image\_ori,vert,mode='same')

for r in range (0,sh[0]):

for c in range (0,sh[1]):

if vert\_img [r][c] < 0:

vert\_img [r][c]= 0

io.imshow (vert\_img, cmap='gray')



forw =[[0,1,2],[-1,0,1],[-2,-1,0]]

diag\_img = signal.convolve2d(image\_ori,forw,mode='same')

for r in range (0,sh[0]):

for c in range (0,sh[1]):

if diag\_img [r][c] < 0:

diag\_img [r][c]= 0

io.imshow (diag\_img, cmap='gray')



back =[[-2,-1,0],[-1,0,1],[0,1,2]]

diag\_img2 = signal.convolve2d(image\_ori,back,mode='same')

for r in range (0,sh[0]):

for c in range (0,sh[1]):

if diag\_img2 [r][c] < 0:

diag\_img2 [r][c]= 0

io.imshow (diag\_img2, cmap='gray')



diagonal\_img = diag\_img + diag\_img2

io.imshow (diagonal\_img, cmap='gray')



edge = [[1,1,1],[1,-8,1],[1,1,1]]

edge\_img = signal.convolve2d(image\_ori,edge,mode='same')

for r in range (0,sh[0]):

for c in range (0,sh[1]):

if edge\_img [r][c] < 0:

edge\_img [r][c]= 0

final\_image = image\_ori + edge\_img

plt.figure (figsize=(10,10))

plt.subplot(1,2,1)

io.imshow (image\_ori, cmap='gray')

plt.title ("Original Diagonal")

plt.subplot(1,2,2)

io.imshow (final\_image, cmap='gray')

plt.title ("Enhanced Diagonal")



CONCLUSIONS:

1. To highlight all the edges of the image, i.e horizontal, vertical and diagonal edges, we use the Sobal Filter.

2. For overall edge enhancement of the image, we use the Laplace Filter, which highlights all the edges of the image

3. For overall enhancement of the image, we add the enhanced image to the original image