BAHRIA UNIVERSITY, ISLAMABAD
Department of Computer Science

CEN 444
Digital Image Processing
Lab Journal 10

Student Name: _M MUNEEB AHMED KIANI

Enrolment No.: 01-135212-063

Title: Edge Detection

Objectives: The purpose of today's lab is to introduce you to the process of edge detection. This lab spotlights the built-in Python IPT functions for different edge and line detection filters.

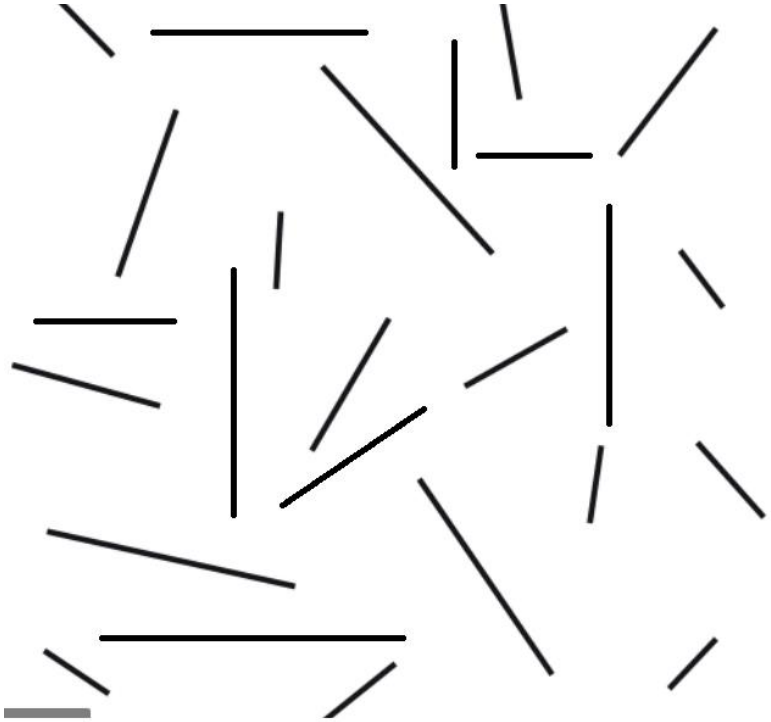
Tools Used: Python

Procedure: Open IDLE and perform the following tasks

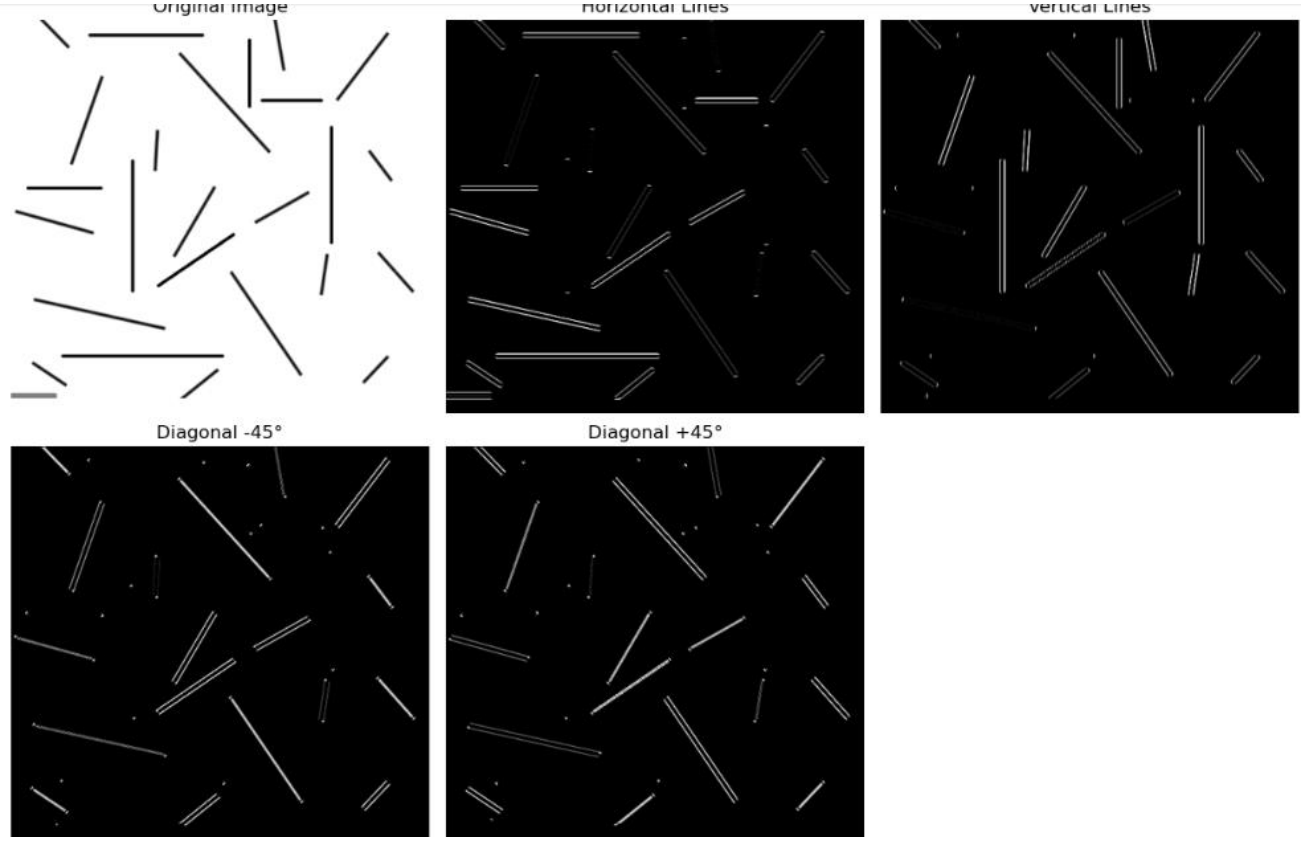
Task 1

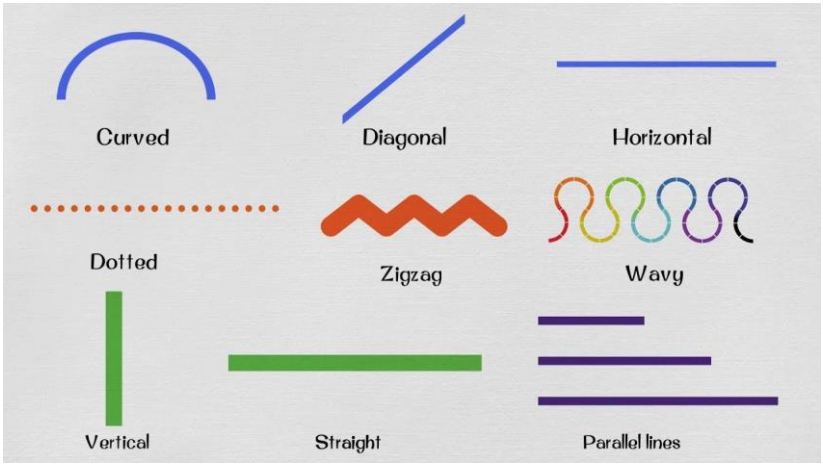
Read the image below, apply the masks to detect horizontal, vertical and diagonal lines and compare the results of different masks. Below are four different line detection filters.

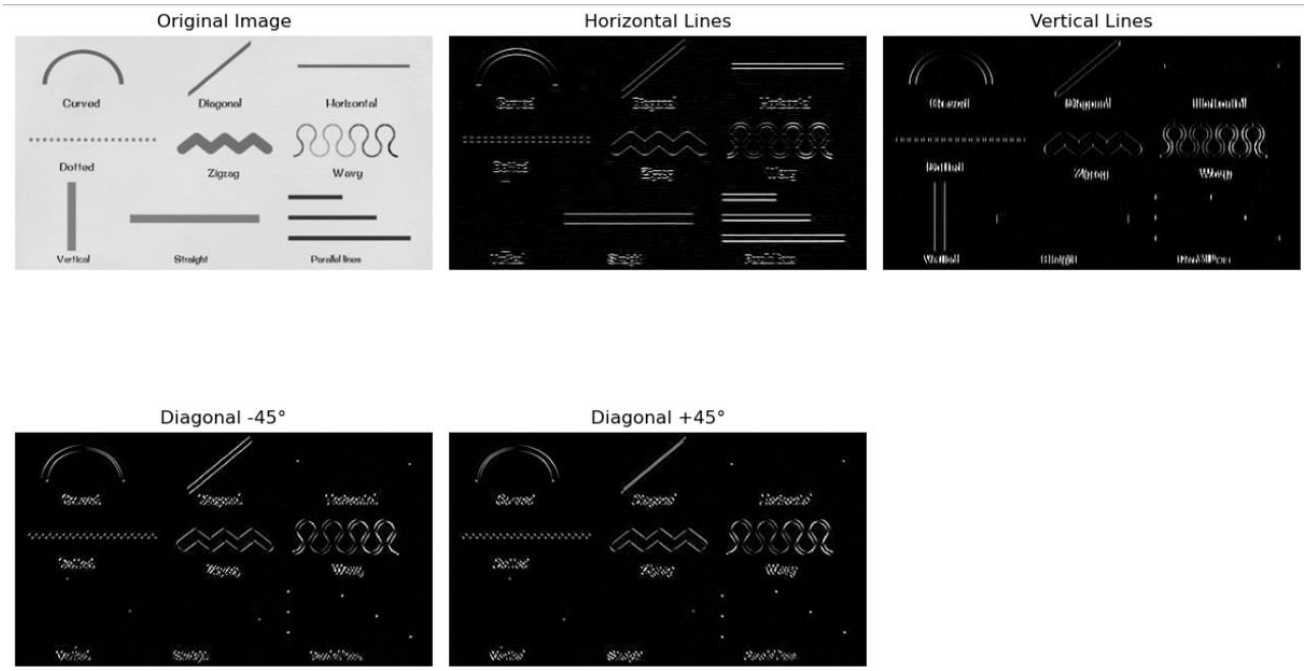
−1 −1 −1	2 −1 −1	−1 2 −1	−1 −1 2
2 2 2	−1 2 −1	−1 2 −1	−1 2 −1
−1 −1 −1	−1 −1 2	−1 2 −1	2 −1 −1
Horizontal	-45°	Vertical	+ 45°



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

img = cv2.imread('Picture2.png', cv2.IMREAD_GRAYSCALE)

horizontal_mask = np.array([[-1, -1, -1], [2, 2, 2], [-1, -1, -1]])
vertical_mask = np.array([[-1, 2, -1], [-1, 2, -1], [-1, 2, -1]])
diagonal_45_mask = np.array([[2, -1, -1], [-1, 2, -1], [-1, -1, 2]])
diagonal_neg45_mask = np.array([[-1, -1, 2], [-1, 2, -1], [2, -1, -1]])

horizontal_lines = cv2.filter2D(img, -1, horizontal_mask)
vertical_lines = cv2.filter2D(img, -1, vertical_mask)
diagonal_45_lines = cv2.filter2D(img, -1, diagonal_45_mask)
diagonal_neg45_lines = cv2.filter2D(img, -1, diagonal_neg45_mask)

titles = ['Original Image', 'Horizontal Lines', 'Vertical Lines', 'Diagonal -45°', 'Diagonal +45°']
images = [img, horizontal_lines, vertical_lines, diagonal_neg45_lines, diagonal_45_lines]

plt.figure(figsize=(12, 8))
for i in range(len(images)):
    plt.subplot(2, 3, i+1)
    plt.imshow(images[i], cmap='gray')
    plt.title(titles[i])
    plt.axis('off')

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


Original Image
Horizontal Lines
Vertical Lines
Diagonal -45°
Diagonal +45°

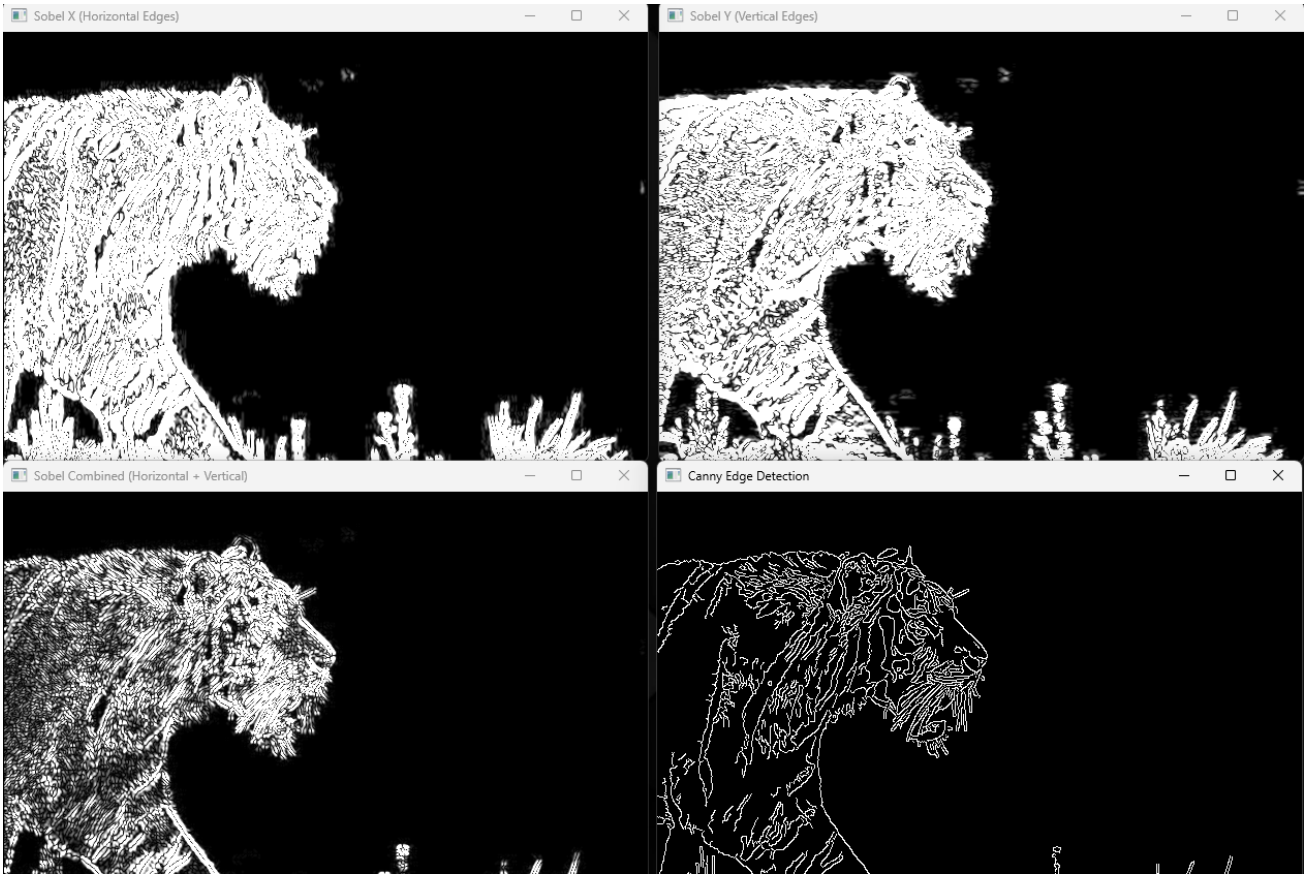

Curved
Diagonal
Horizontal
Dotted
Zigzag
Wavy
Vertical
Straight
Parallel lines



Task 2

Find the horizontal and vertical edges in the following picture. Display both horizontal, vertical and combined edges. Use a sobel filter. Implement both using the filter and the built-in function.

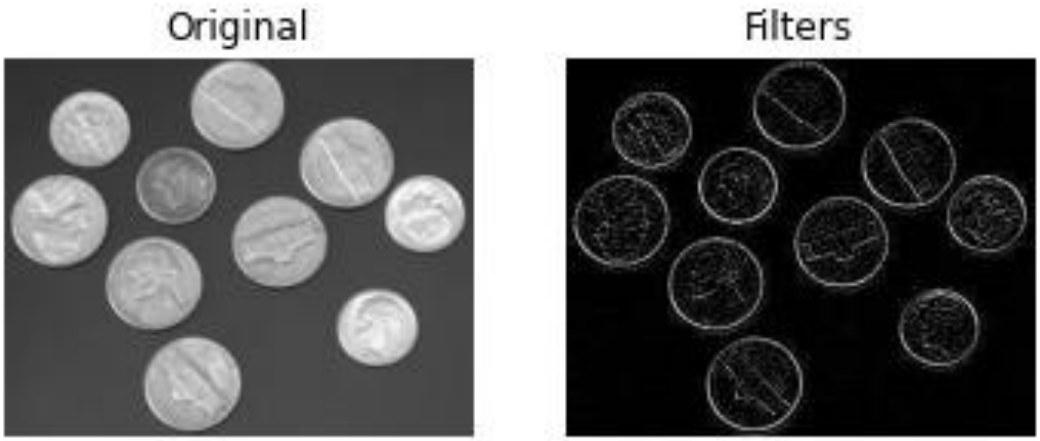
```python
import cv2
import numpy as np

img = cv2.imread('Picture4.jpg')

img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img_blur = cv2.GaussianBlur(img_gray, (3, 3), 0)
sobelx = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=0, ksize=5)
sobely = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=0, dy=1, ksize=5)
sobelxy = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5)

sobelx = cv2.convertScaleAbs(sobelx)
sobely = cv2.convertScaleAbs(sobely)
sobelxy = cv2.convertScaleAbs(sobelxy)

cv2.imshow('Original', img)
cv2.imshow('Sobel X (Horizontal Edges)', sobelx)
cv2.imshow('Sobel Y (Vertical Edges)', sobely)
cv2.imshow('Sobel Combined (Horizontal + Vertical)', sobelxy)

edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200)
cv2.imshow('Canny Edge Detection', edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



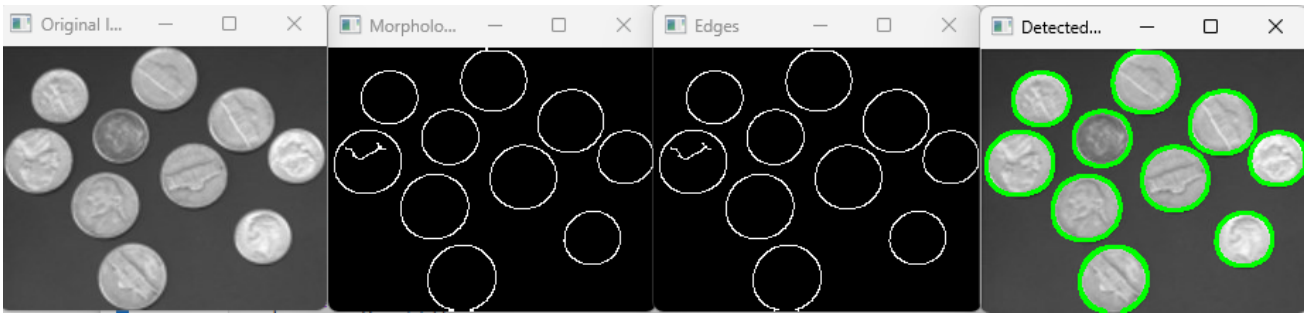
Task 3:

Read the image ‘coin’, you have to get the required output image shown in figure below. Analyze the input image and make the decision yourself what to do to get the required output.

Hint: Choose kernel/mask values yourself to get required output.



```python
import cv2
import numpy as np

img = cv2.imread('coins.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (7, 7), 0)
edges = cv2.Canny(blurred, threshold1=50, threshold2=150)
kernel = np.ones((3, 3), np.uint8)
morph = cv2.morphologyEx(edges, cv2.MORPH_CLOSE, kernel)
contours, _ = cv2.findContours(morph, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
output = img.copy()
cv2.drawContours(output, contours, -1, (0, 255, 0), 2)
cv2.imshow('Original Image', img)
cv2.imshow('Morphological Transformation', morph)
cv2.imshow('Edges', edges)
cv2.imshow('Detected Coins', output)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Submission Date