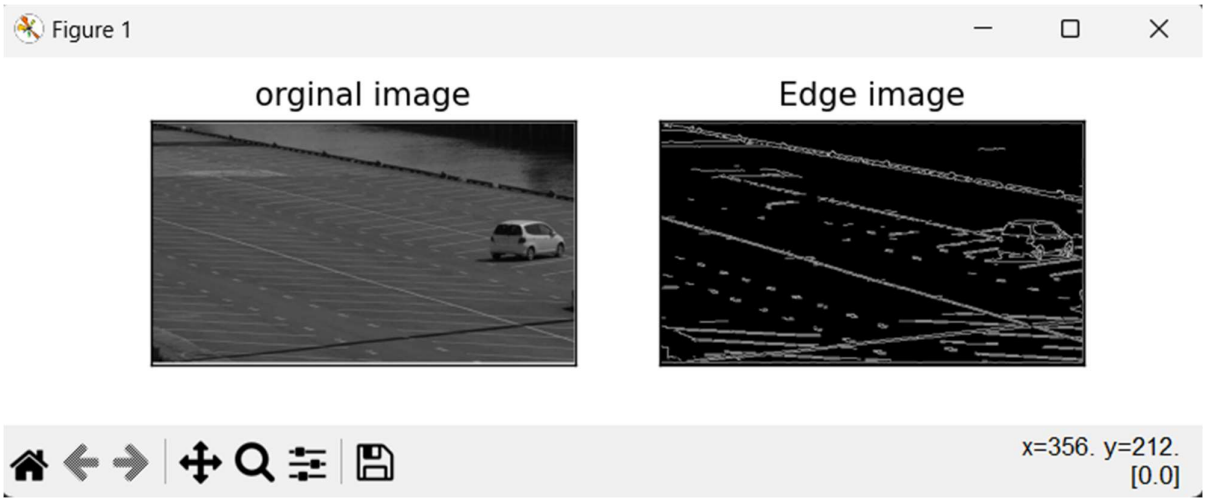
Practical 7:-

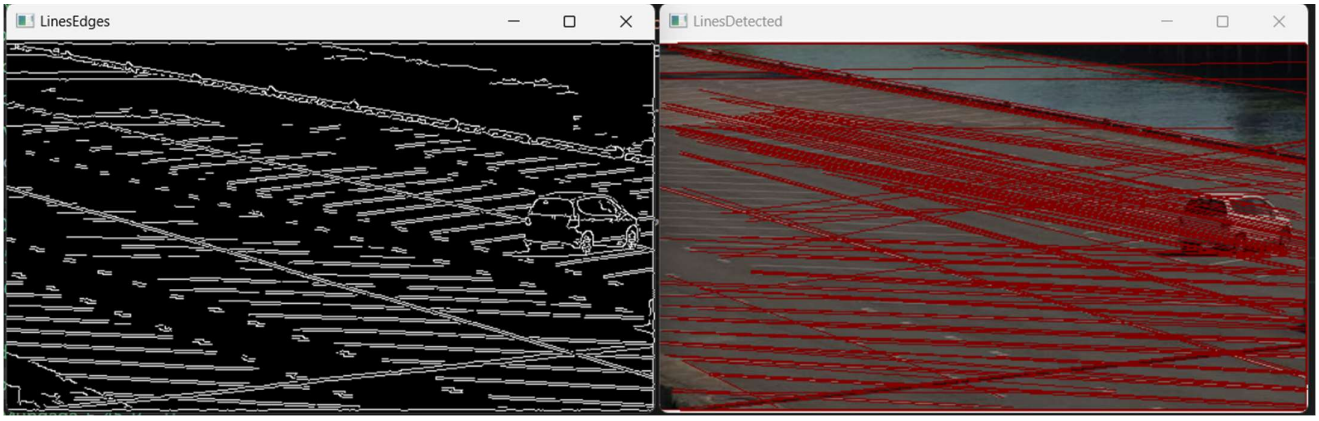
Canny Edge

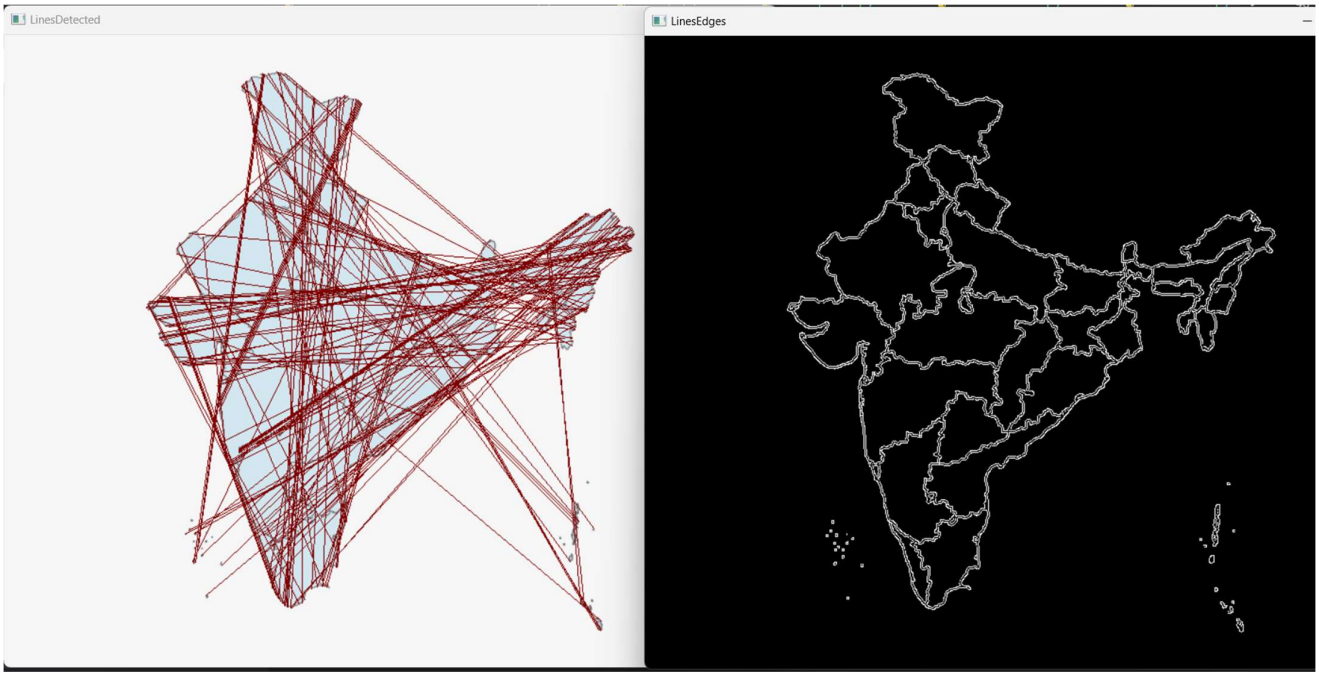
```
import cv2
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
img=cv2.imread("assets/tiger.jpg",cv2.IMREAD_GRAYSCALE)
#cv2.imshow('original',img)
# assert img is not None: "file could not read"
edges=cv2.Canny(img,100,200)
#print(img.shape)
plt.subplot(121), plt.imshow(img,cmap='gray')
plt.title('orginal image'),plt.xticks([]),plt.yticks([])
plt.subplot(122),plt.imshow(edges,cmap='gray')
plt.title('Edge image'),plt.xticks([]),plt.yticks([])
plt.show()
```



Hough Filter:-

```
import cv2
import numpy as np
img=cv2.imread('assets/parking_lot.jpg')
gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
edges=cv2.Canny(gray,75,150)
lines=cv2.HoughLinesP(edges,1,np.pi/180,30,maxLineGap=250)

for line in lines:
    x1,y1,x2,y2=line[0]
    cv2.line(img,(x1,y1),(x2,y2),(0,0,128,1))
cv2.imshow("LinesEdges ",edges)
cv2.imshow("LinesDetected ",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```





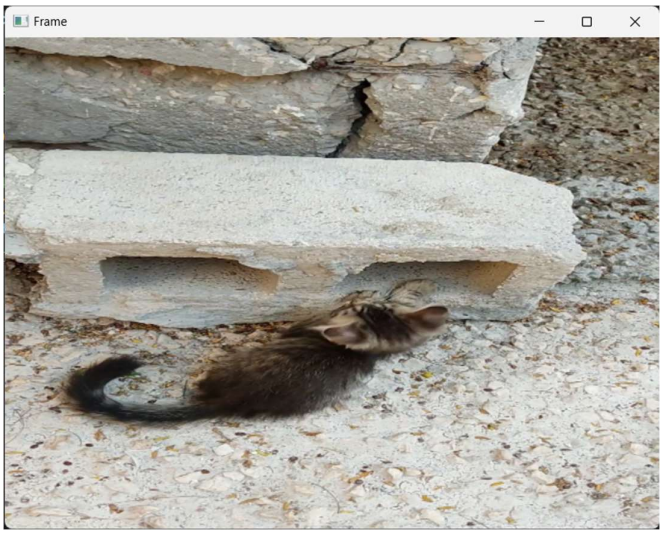
Practical 8
Video Properties:-

```
# importing cv2
import cv2

# For Video File
capture=cv2.VideoCapture(r"assets\video.mp4")

# For webcam
#capture = cv2.VideoCapture(0)

# showing values of the properties
print("CV_CAP_PROP_FRAME_WIDTH: '{}'".format(capture.get(cv2.CAP_PROP_FRAME_WIDTH)))
print("CV_CAP_PROP_FRAME_HEIGHT : '{}'".format(capture.get(cv2.CAP_PROP_FRAME_HEIGHT)))
print("CAP_PROP_FPS : '{}'".format(capture.get(cv2.CAP_PROP_FPS)))
print("CAP_PROP_POS_MSEC : '{}'".format(capture.get(cv2.CAP_PROP_POS_MSEC)))
print("CAP_PROP_FRAME_COUNT  : '{}'".format(capture.get(cv2.CAP_PROP_FRAME_COUNT)))
print("CAP_PROP_BRIGHTNESS : '{}'".format(capture.get(cv2.CAP_PROP_BRIGHTNESS)))
print("CAP_PROP_CONTRAST : '{}'".format(capture.get(cv2.CAP_PROP_CONTRAST)))
print("CAP_PROP_SATURATION : '{}'".format(capture.get(cv2.CAP_PROP_SATURATION)))
print("CAP_PROP_HUE : '{}'".format(capture.get(cv2.CAP_PROP_HUE)))
print("CAP_PROP_GAIN  : '{}'".format(capture.get(cv2.CAP_PROP_GAIN)))
print("CAP_PROP_CONVERT_RGB : '{}'".format(capture.get(cv2.CAP_PROP_CONVERT_RGB)))

# release window
capture.release()
cv2.destroyAllWindows()
```

```
PROBLEMS    OUTPUT    TERMINAL    SQL CONSOL

CV_CAP_PROP_FRAME_WIDTH: '1080.0'
CV_CAP_PROP_FRAME_HEIGHT : '1920.0'
CAP_PROP_FPS : '29.97002997002997'
CAP_PROP_POS_MSEC : '0.0'
CAP_PROP_FRAME_COUNT  : '467.0'
CAP_PROP_BRIGHTNESS : '0.0'
CAP_PROP_CONTRAST : '0.0'
CAP_PROP_SATURATION : '0.0'
CAP_PROP_HUE : '0.0'
CAP_PROP_GAIN  : '0.0'
CAP_PROP_CONVERT_RGB : '0.0'
PS E:\6th_Sem\FDIP_lab>
```

VideoProcess:-

```
import cv2
# Create a video capture object, in this case we are reading the video from a file
vid_capture = cv2.VideoCapture(r"assets\video.mp4")
if (vid_capture.isOpened() == False):
    print("Error opening the video file")
# Read fps and frame count
else:
    # Get frame rate information
    # You can replace 5 with CAP_PROP_FPS as well, they are enumerations
    fps = vid_capture.get(5)
    print('Frames per second : ', fps, 'FPS')
    # Get frame count
    # You can replace 7 with CAP_PROP_FRAME_COUNT as well, they are enumerations
    frame_count = vid_capture.get(7)
    print('Frame count : ', frame_count)
while (vid_capture.isOpened()):
    # vid_capture.read() methods returns a tuple, first element is a bool
    # and the second is frame
    ret, frame = vid_capture.read()
    if ret == True:
        # Resize the frame to 640x480
        resized_frame = cv2.resize(frame, (640, 480))
        cv2.imshow('Frame', resized_frame)
        # 20 is in milliseconds, try to increase the value, say 50 and observe
        key = cv2.waitKey(20)
        if key == ord('q'):
            break
    else:
        break
# Release the video capture object
vid_capture.release()
cv2.destroyAllWindows()
```




```
PROBLEMS  OUTPUT  TERMINAL  SQL CONSOLE  DEBUG CONSOLE  COMMENTS

PS E:\6th_Sem\FDIP_lab> python -u "e:\6th_Sem\FDIP_lab\Practical_8\videoProcess.py"
Frames per second :  29.97002997002997 FPS
Frame count :  467.0
PS E:\6th_Sem\FDIP_lab>
```

Video Save:-

```
# Python program to save a video using OpenCV
import cv2
# Create an object to read from camera
video = cv2.VideoCapture(r"assets\video.mp4")

# We need to check if camera is opened previously or not
if (video.isOpened() == False):
    print("Error reading video file")

# We need to set resolutions. so, convert them from float to integer.
frame_width = int(video.get(3))
frame_height = int(video.get(4))

size = (frame_width, frame_height)

# Below VideoWriter object will create a frame of above defined The output
# is stored in 'filename.avi' file.
result = cv2.VideoWriter('filename.avi',cv2.VideoWriter_fourcc(*'MJPG'),10, size)

while (True):
    ret, frame = video.read()
    if ret == True:
        # Write the frame into the
        # file 'filename.avi'
        result.write(frame)
        # Display the frame saved in the file
        frame = cv2.resize(frame, (640, 480))
        cv2.imshow('Frame', frame)
        # Press S on keyboard to stop the process
        if cv2.waitKey(1) & 0xFF == ord('s'):
            break
    # Break the loop
    else:
        break
# When everything done, release the video capture and video write objects
video.release()
result.release()
# Closes all the frames
cv2.destroyAllWindows()

print("The video was successfully saved")
```

```
PS E:\6th_Sem\FDIP_lab> python -u "e:\6th_Sem\FDIP_lab\Practical_8\videoSave.py"
The video was successfully saved
PS E:\6th_Sem\FDIP_lab>
```