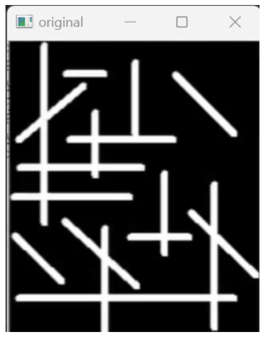
**Aim:**
**To study and perform line and edge detection methods using opencv python**

1. **To detect horizontal and vertical lines in an image**
2. **To illustrate Houghline method for line detection**
3. **To illustrate Canny edge detection algorithm**

1) **To detect Horizontal and Vertical Lines in an Image**

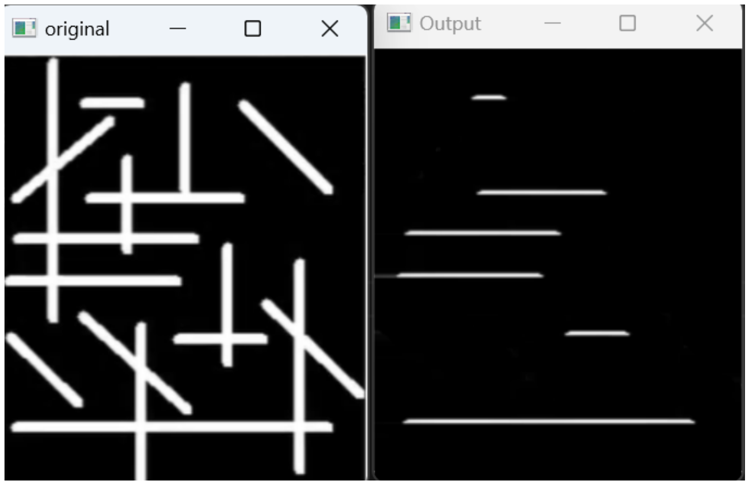
**Input Image:**



**Code for Horizontal:**

```
import cv2

import numpy as np

img=cv2.imread('lines.png',0)
cv2.imshow('original',img)
kernel=np.ones((3,10),np.uint8)
horizontalLines=cv2.erode(img,kernel,iterations=2)
cv2.imshow('Output',horizontalLines)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

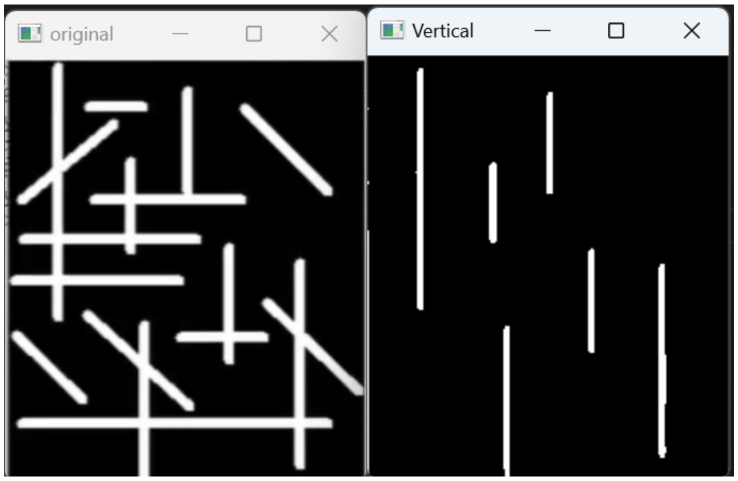
**Output**:



**Code for Vertical:**

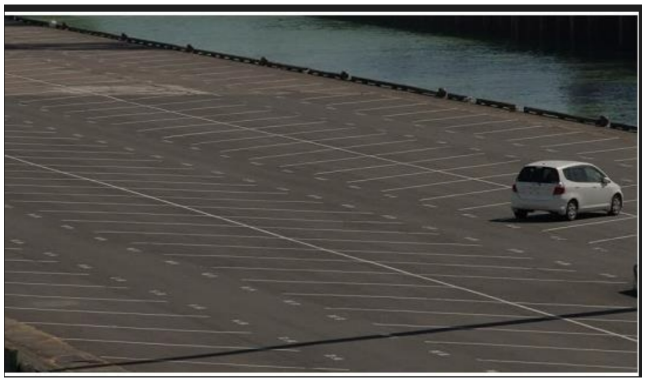
```
import cv2
img=cv2.imread('lines.png',0)
import numpy as np

binr = cv2.threshold(img,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)[1]
kernel = np.ones((11,3),np.uint8)
verticalLines = cv2.erode(binr, kernel, iterations=1)
print('\nOnly Vertical Lines Image \n')
cv2.imshow('original',img)
cv2.imshow("Vertical",verticalLines)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**Output:**



## 2) To illustrate Houghline method for line detection

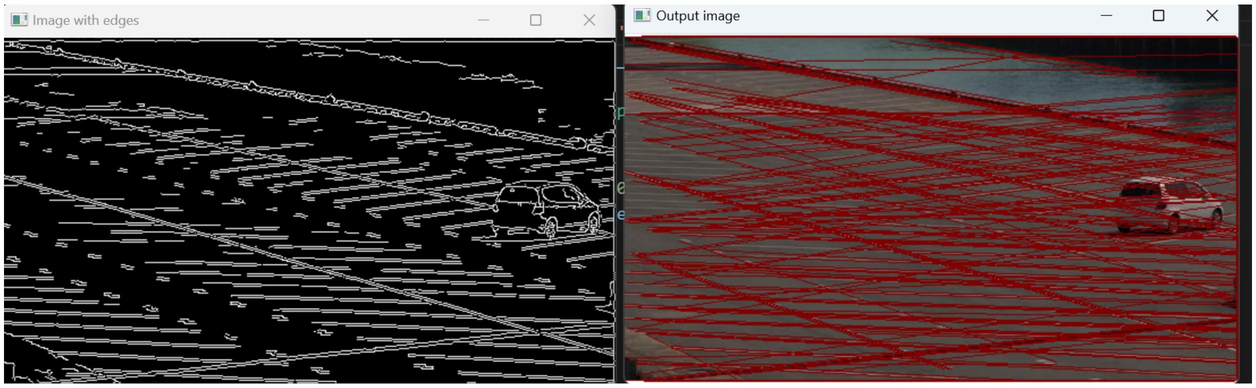
**Input Image:**



**Code:**

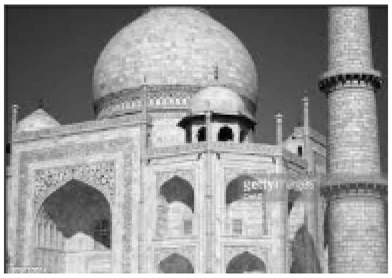
```
import cv2
import numpy as np
img = cv2.imread('parking_lot.jpg')
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(gray,75,150)
lines = cv2.HoughLinesP(edges,1,np.pi/180,20,maxLineGap=250)
for line in lines:
 x1,y1,x2,y2 = line[0]
 cv2.line(img,(x1,y1),(x2,y2),(0,0,128),1)
cv2.imshow("Image with edges",edges)
cv2.imshow("Output image",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

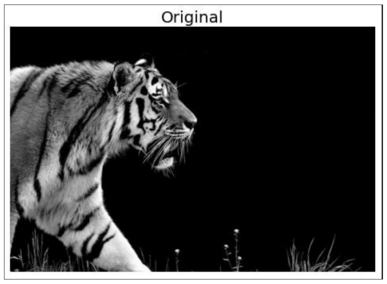
**Output:**

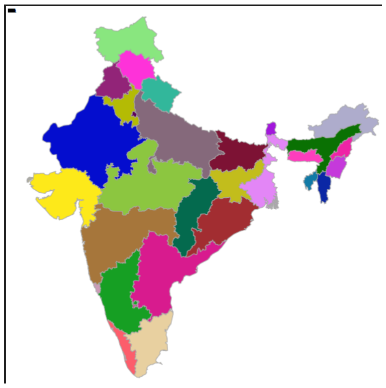


**3) To illustrate Canny edge detection algorithm**

**Input Image:**





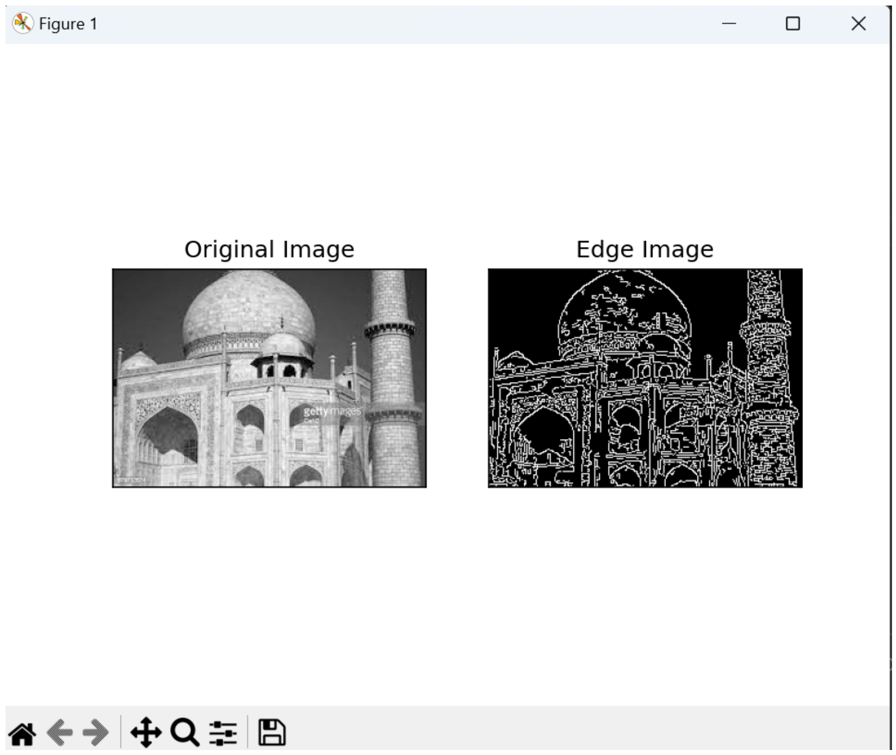


**Code:**

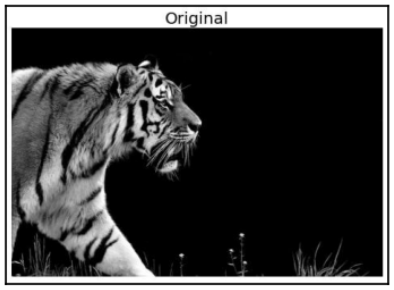
```
import cv2
import numpy as np
```

```
from matplotlib import pyplot as plt
img = cv2.imread('map2.png',cv2.IMREAD_GRAYSCALE)
edges = cv2.Canny(img,100,200)
plt.subplot(121)
plt.imshow(img,cmap = 'gray')
plt.title('Original Image')
plt.xticks([])
plt.yticks([])
plt.subplot(122)
plt.imshow(edges,cmap='gray')
plt.title('Edge Image')
plt.xticks([])
plt.yticks([])
plt.show()
```

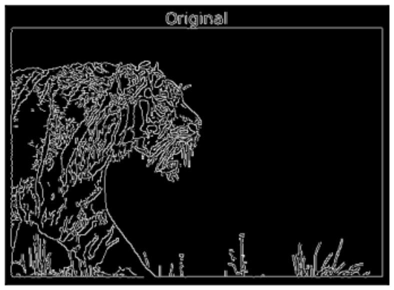
**Output Image:**



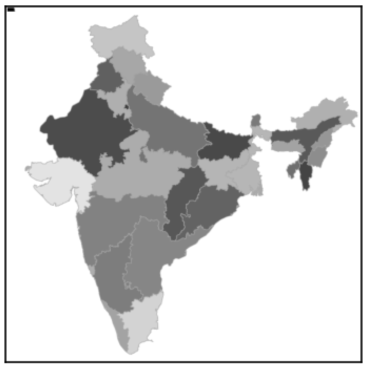
Original Image



Edge Image



Original Image



Edge Image