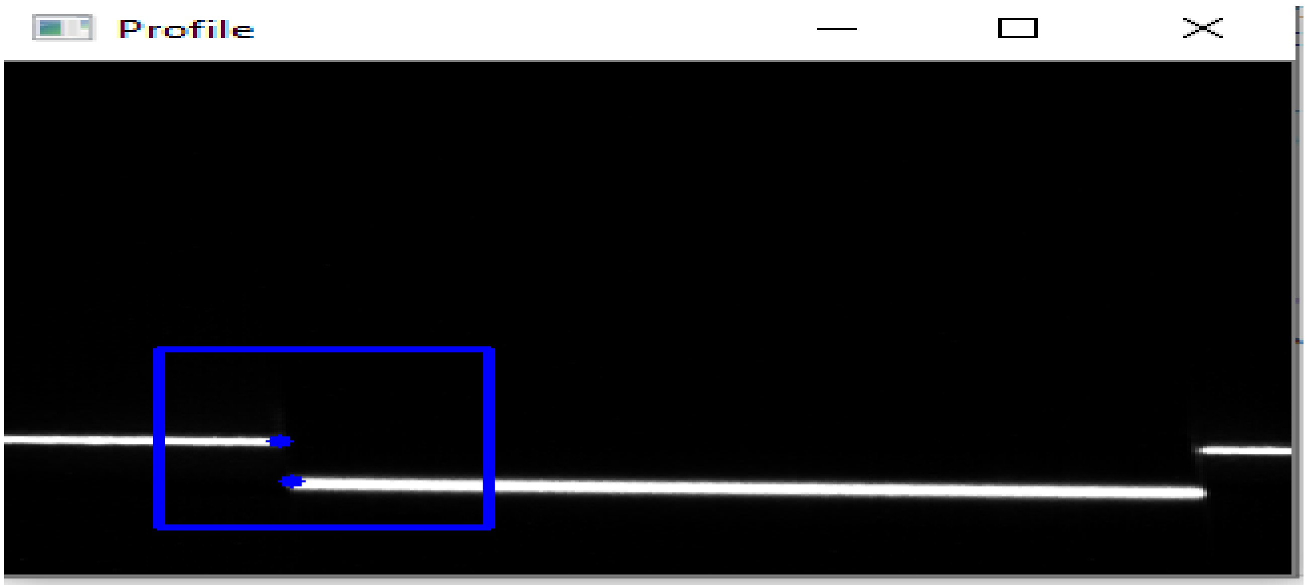

```
68  :  189
71  :  209
```

```
import cv2

import numpy as np

%matplotlib inline

import matplotlib.pyplot as plt

import matplotlib.image as mpimg


# Kullanılacak görüntünün dosya ismini kullanıcıya sormak

print('Dosya: ')

filename = input()

img1 = cv2.imread(filename)

img = cv2.resize(img1, (320, 256))

# Mouse ile kullanıcın  alan seçmesine izin vermek

ix = -1

iy = -1

drawing = False

selected = False
```

```
def draw_reactangle_with_drag(event, x, y, flags, param):

    global ix, iy, drawing, img

    if event == cv2.EVENT_LBUTTONDOWN:

        drawing = True

        ix = x

        iy = y



    elif event == cv2.EVENT_MOUSEMOVE:

        if drawing == True:

            img2 = cv2.resize(img1, (320, 256))

            cv2.rectangle(img2, pt1=(ix,iy), pt2=(x, y),color=(255,0,0),thickness=2)

            img = img2


    elif event == cv2.EVENT_LBUTTONUP:

        drawing = False

        img2 = cv2.resize(img1, (320, 256))

        cv2.rectangle(img2, pt1=(ix,iy), pt2=(x, y),color=(255,0,0),thickness=2)

        img = img2


        # Crop

        crop_img = img2[iy:y, ix:x]



        # Kenar bulma algoritması ile köşeleri bulmak mavi ile işaretlemek

        gray = cv2.cvtColor(crop_img, cv2.COLOR_BGR2GRAY)
        corners = cv2.goodFeaturesToTrack(gray, 27, 0.3, 10)
        corners = np.int0(corners)

        # koseleri cizdir
        for i in corners:
            xc, yc = i.ravel()

            height, width, channels = crop_img.shape

            # cercevede bulunanlari cizdirme
            if xc<5 or yc<5 or xc>width-5:
                continue

            # koselere yuvarlak koy
            cv2.circle(img, (ix+xc, iy+yc), 3, 255, -1)

            # koselerin koordinatlari
            print(ix+xc,':',iy+yc)

        plt.imshow(img)

cv2.namedWindow(winname= "Profile")
cv2.setMouseCallback("Profile", draw_reactangle_with_drag)

while True:
    cv2.imshow("Profile", img)
    if cv2.waitKey(10) == 27:
        break
cv2.destroyAllWindows()
```