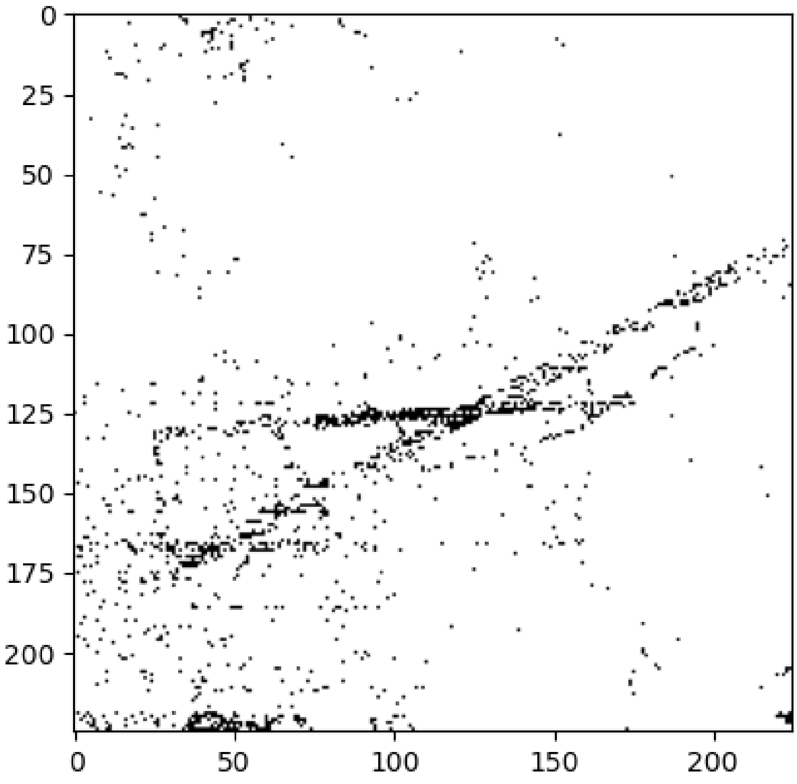
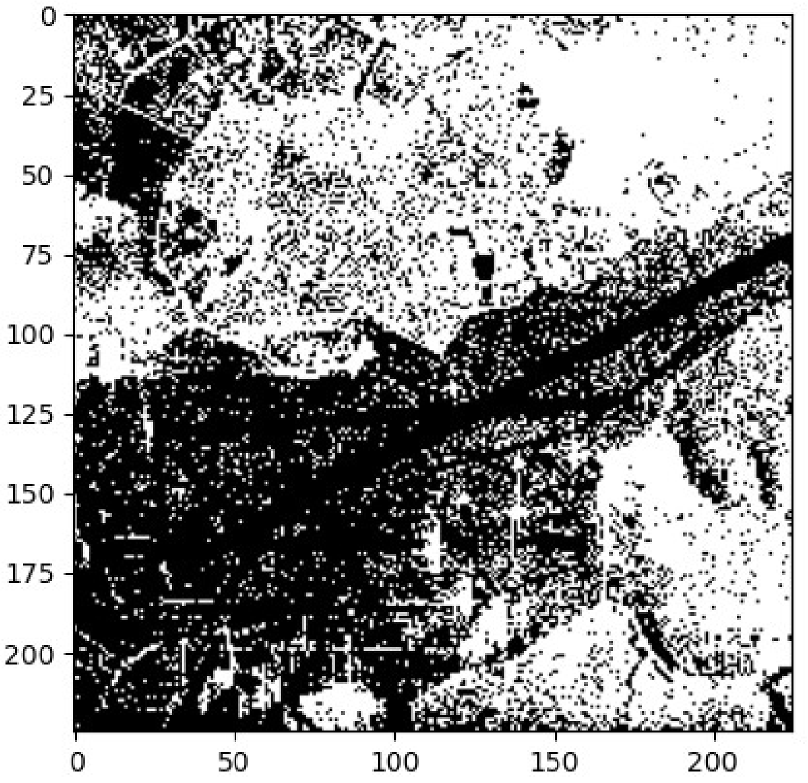
```
import numpy as np
import cv2
import matplotlib.pyplot as plt

image = cv2.imread('sar_3.jpg')
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
eq_gray = cv2.equalizeHist(cv2.cvtColor(image, cv2.COLOR_BGR2GRAY))
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# точечная бинаризация

import copy

bin_img = copy.deepcopy(image_gray)
T  = 40
bin_img[image_gray < T] = 0
bin_img[image_gray >= T] = 255

plt.imshow(bin_img, cmap="gray")

<matplotlib.image.AxesImage at 0x790ca20>
```

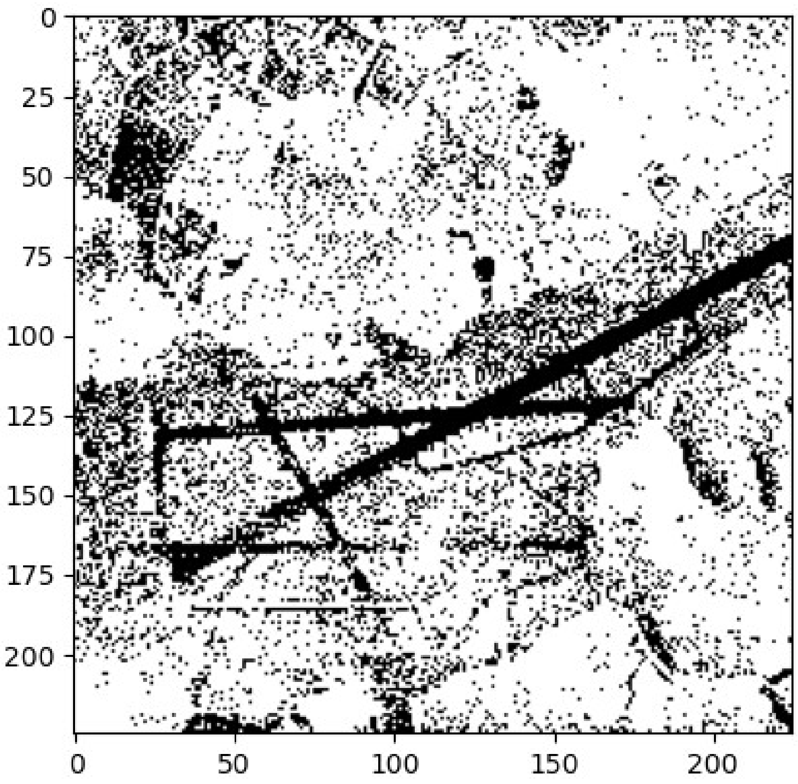


```
# бинаризация отсу

_,th2 =
cv2.threshold(image_gray,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
```

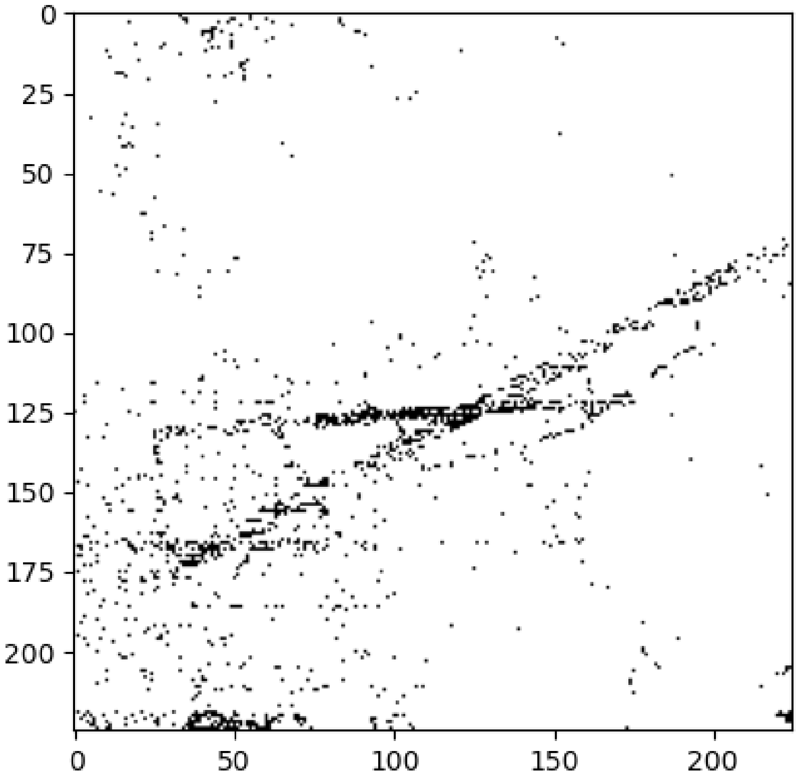
```
plt.imshow(th2, cmap="gray")

<matplotlib.image.AxesImage at 0x75bdb60>
```



```
# адаптивная бинаризация

th3 =
cv2.adaptiveThreshold(image_gray,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv
2.THRESH_BINARY,71,21)
th31 =
cv2.adaptiveThreshold(eq_gray,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRES
H_BINARY,101,45)

#plt.imshow(th3, cmap="gray")

plt.imshow(th31, cmap="gray")

<matplotlib.image.AxesImage at 0x761a1c0>
```



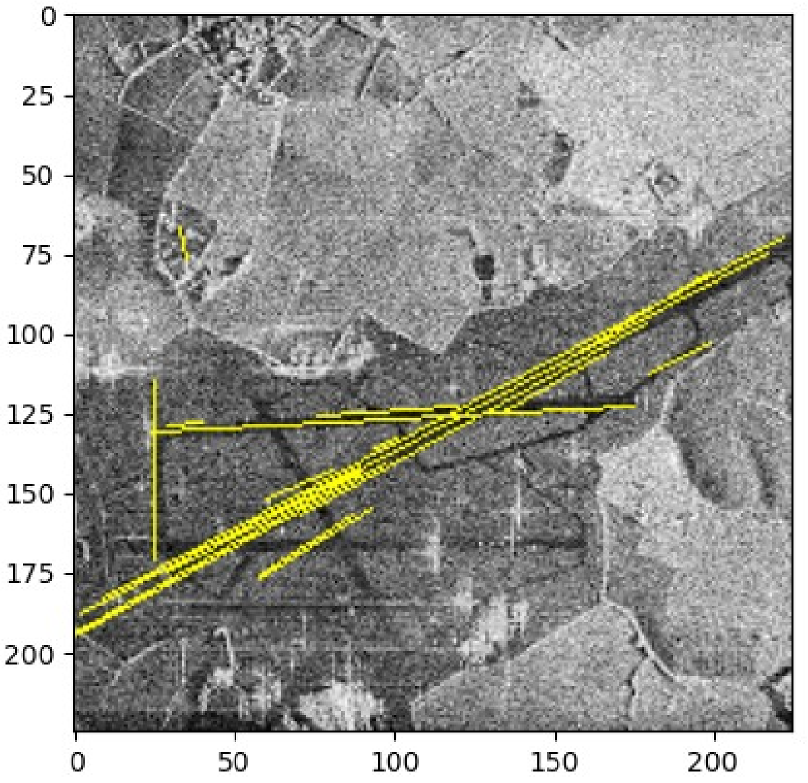
```
# выделение дороги, подсчёт протяженности (длины отрезков)

plt.imshow(bin_img, cmap="gray")

<matplotlib.image.AxesImage at 0x77f07a0>
```



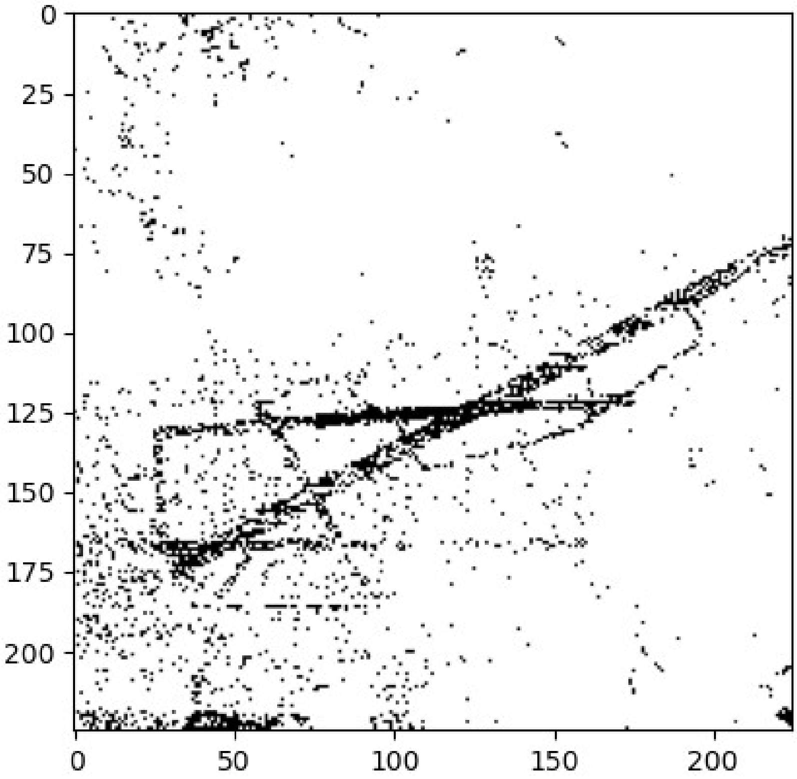
```
image = cv2.imread('sar_3.jpg')
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
eq_gray = cv2.equalizeHist(cv2.cvtColor(image, cv2.COLOR_BGR2GRAY))
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

edges = cv2.Canny(bin_img,0,1,apertureSize = 3)
minLineLength = 10
tlines =
cv2.HoughLinesP(edges,1,np.pi/180,50,minLineLength,maxLineGap=8)

maxum = 0 # наибольший отрезок дороги

for i in range(len(tlines)):
    x1 = tlines[i][0][0]
    y1 = tlines[i][0][1]
    x2 = tlines[i][0][2]
    y2 = tlines[i][0][3]
    cv2.line(image,(x1,y1),(x2,y2),(255,255,0),1)
    tlen = np.sqrt((x2-x1)**2+(y2-y1)**2)
    if (tlen > maxum):
        maxum = tlen

print(maxum)
```

```
241.11822826157297
```

```
plt.imshow(image, cmap="gray")

<matplotlib.image.AxesImage at 0x95b9ce8>
```

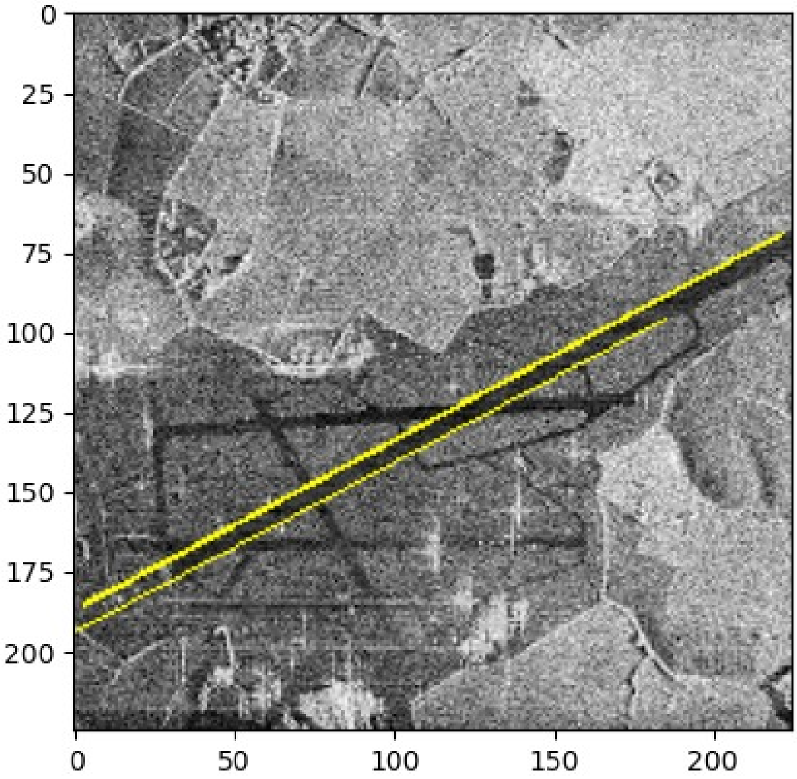


```
image1 = cv2.imread('sar_3.jpg')
image1 = cv2.cvtColor(image1, cv2.COLOR_RGB2BGR)
eq_gray1 = cv2.equalizeHist(cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY))
image_gray1 = cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)

import copy

bin_img1 = copy.deepcopy(eq_gray1)
T  = 16
bin_img1[eq_gray1 < T] = 0
bin_img1[eq_gray1 >= T] = 255

plt.imshow(bin_img1, cmap="gray")

<matplotlib.image.AxesImage at 0x9605b60>
```



```
edges1 = cv2.Canny(bin_img1,0,1,apertureSize = 5)
minLineLength1 = 10
tlines1 =
cv2.HoughLinesP(edges1,1,np.pi/180,100,minLineLength,maxLineGap=10)

for i in range(len(tlines1)):
    x1 = tlines1[i][0][0]
    y1 = tlines1[i][0][1]
    x2 = tlines1[i][0][2]
    y2 = tlines1[i][0][3]
    #print(x1, y1, x2, y2)
    cv2.line(image1,(x1,y1),(x2,y2),(255,255,0),1)

plt.imshow(image1, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x9235540>
```


0
25
50
75
100
125
150
175
200
0
50
100
150
200