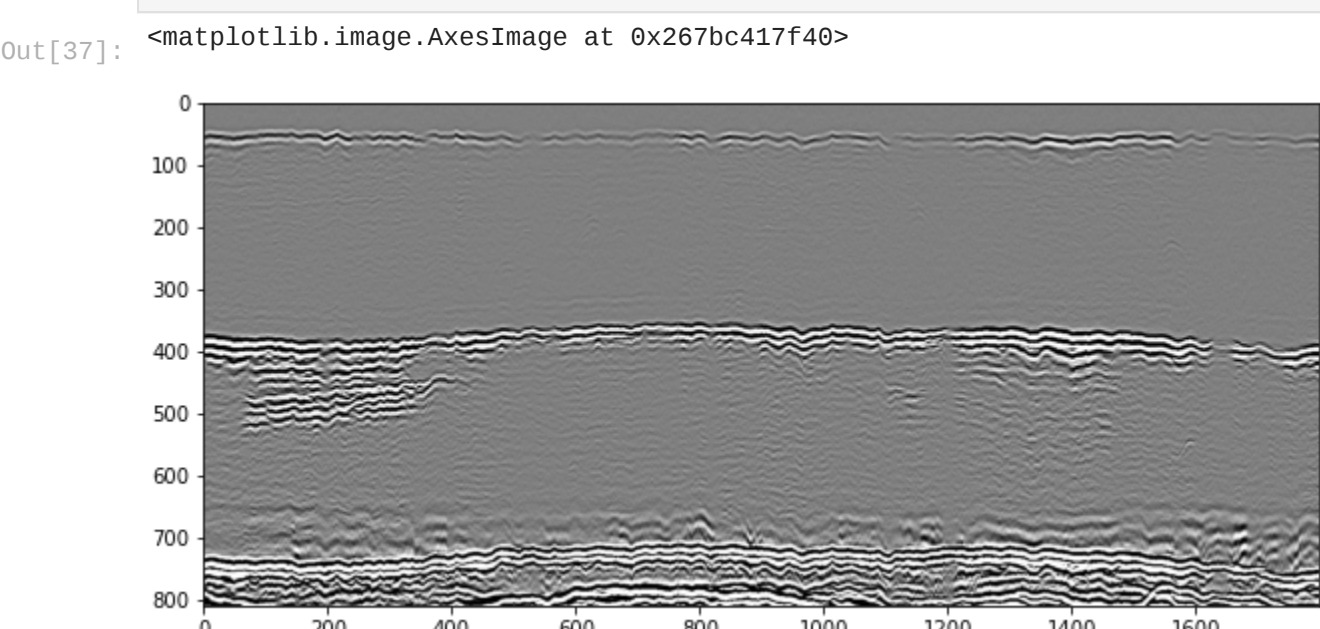


```
In [35]: import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (10,15)
```

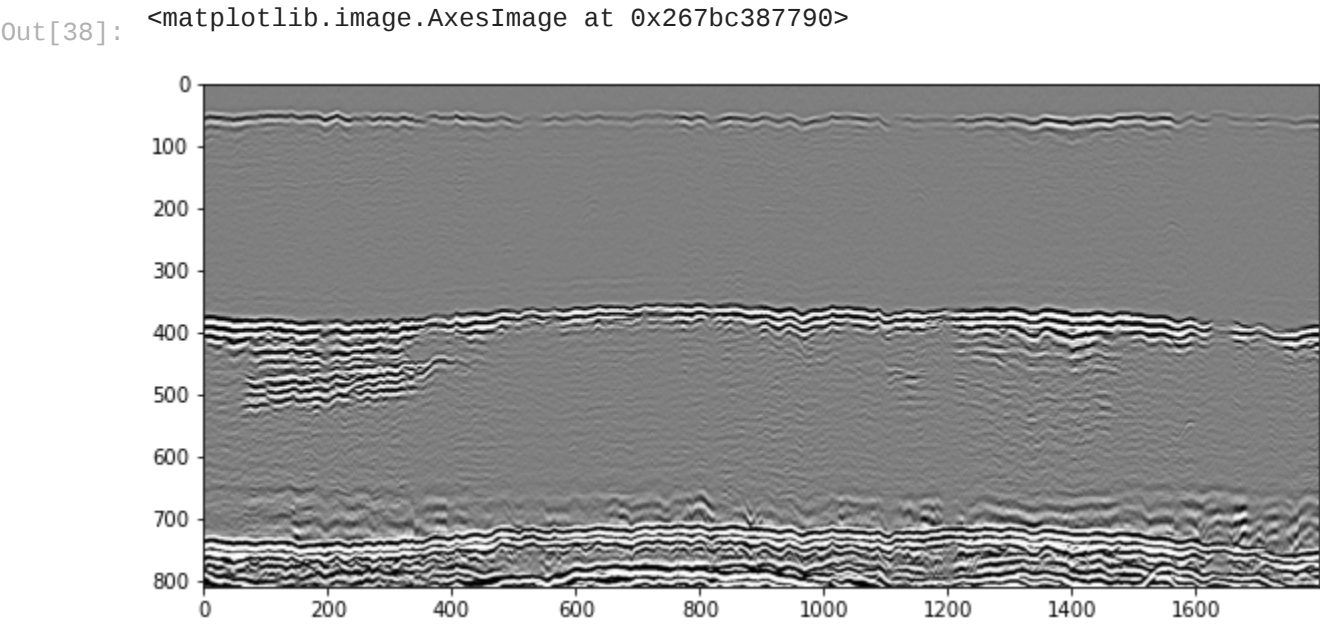
```
In [36]: image = cv.imread(r"C:\Users\DELL\Desktop\L&T 30\S848756-CW-501-SP0T-37-41-70deg-90,pcs-5mhz -CC -Repair.scn.png",0)
```

```
In [37]: plt.imshow(image,"gray")
```

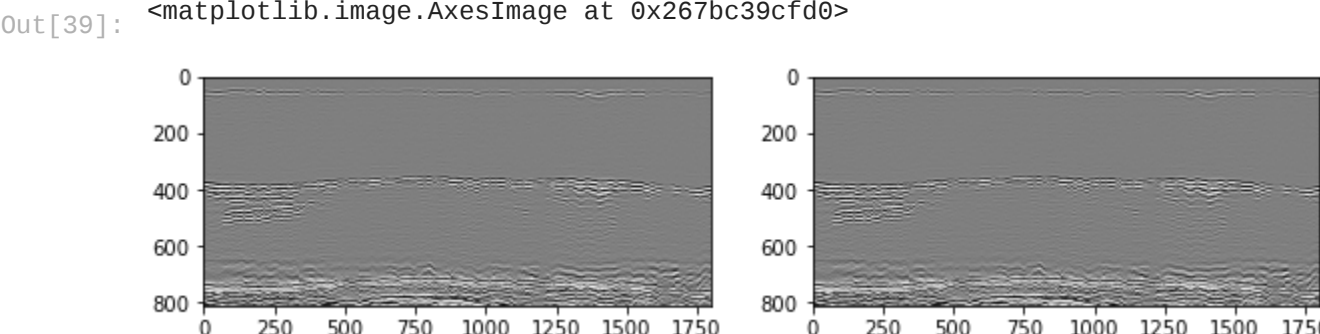


Applying median blur to the image

```
In [38]: median_blur = cv.medianBlur(image , 15)
plt.imshow(image,"gray")
```

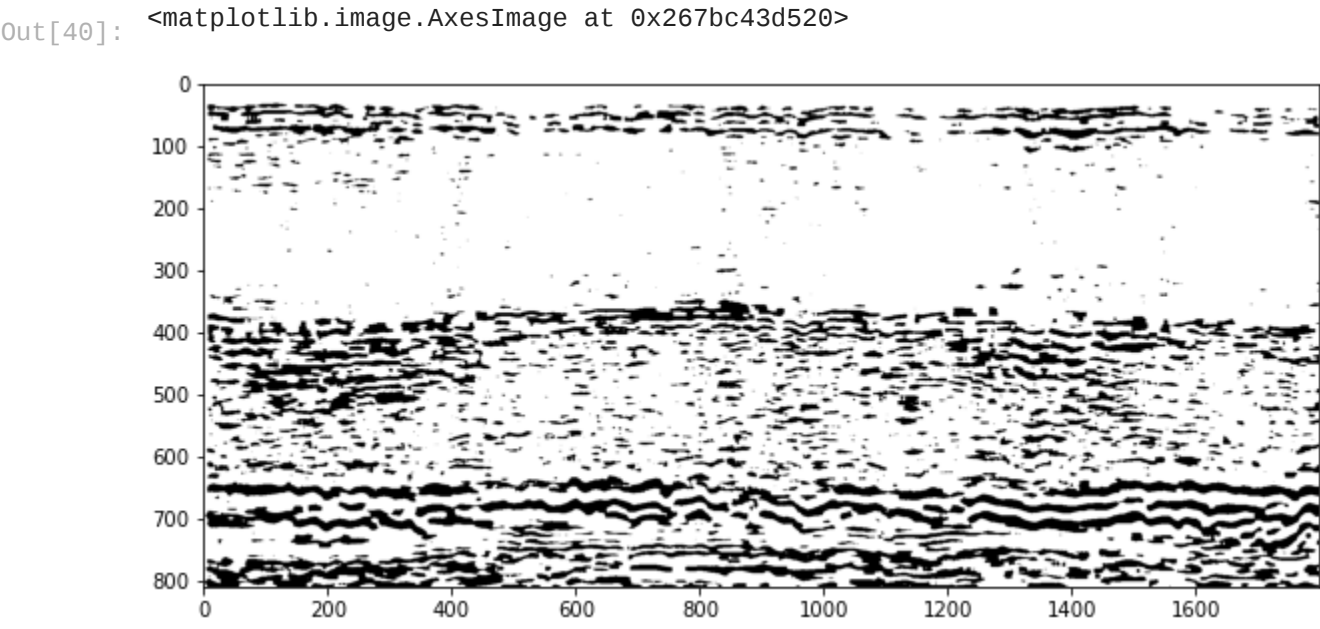


```
In [39]: plt.subplot(121)
plt.imshow(image,"gray")
plt.subplot(122)
median_blur = cv.medianBlur(image , 15)
plt.imshow(image,"gray")
```



Applying the Binary Thresholding

```
In [40]: ret, image_thresh = cv.threshold(median_blur , 125 , 255 , cv.THRESH_BINARY)
plt.imshow(image_thresh,"gray")
```



Finding contours

```
In [41]: contours,hierarchy = cv.findContours(image_thresh.copy() , cv.RETR_CCOMP , cv.CHAIN_APPROX_SIMPLE)
```

Drawing contours.

```
In [42]: image_blank = np.zeros((810,1000) , dtype = "uint8")
```

```
In [43]: image.shape
```

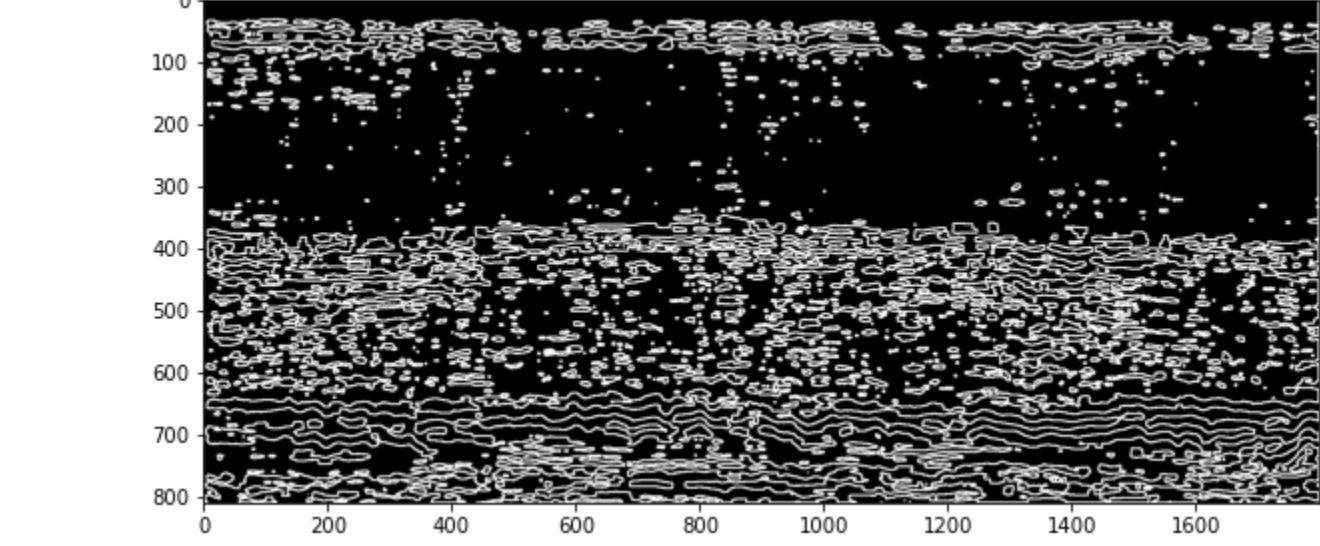
Out[43]: (810, 1000)

```
In [44]: image_blank.shape
```

Out[44]: (810, 1000)

```
In [45]: colored_contours = cv.drawContours(image_blank , contours , -1 , (255,0,0) , 2)
```

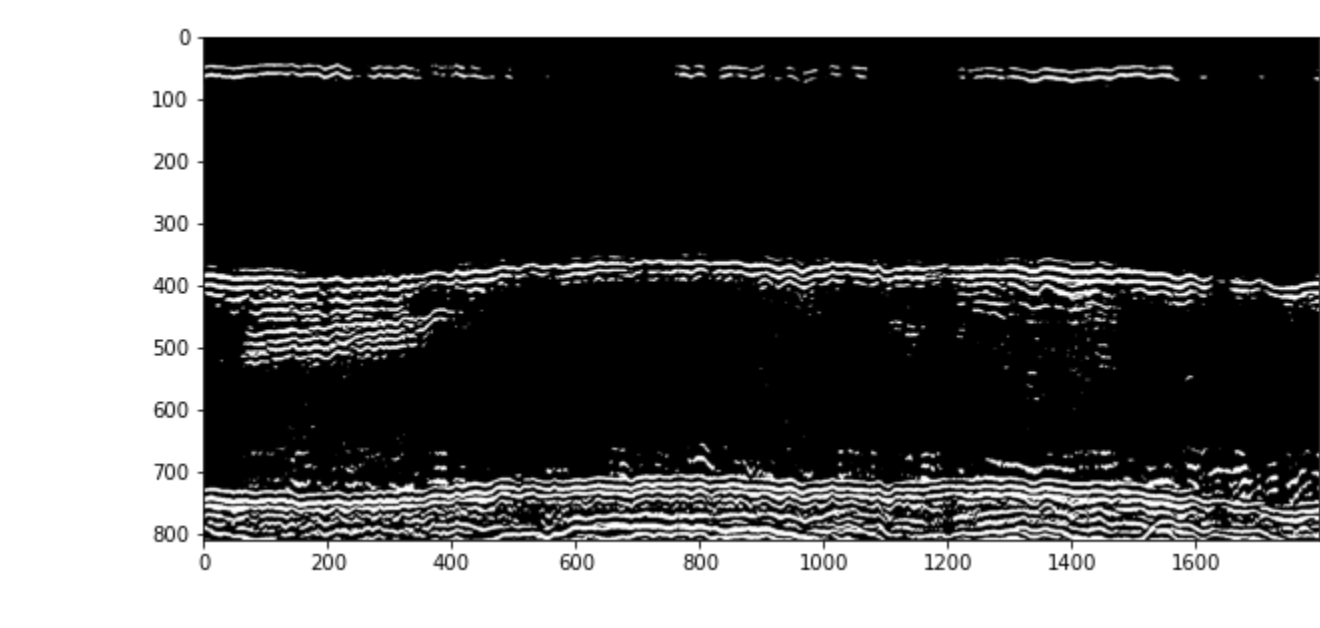
```
In [46]: plt.imshow(colored_contours,"gray")
```



Applying Otsu thresholding to the blurred image.

```
In [47]: ret2 , thresh_otsu = cv.threshold(image , 0 , 255 , cv.THRESH_BINARY+cv.THRESH_OTSU)
```

```
In [48]: plt.imshow(thresh_otsu,"gray")
```



Noise removal (Morphological operations)

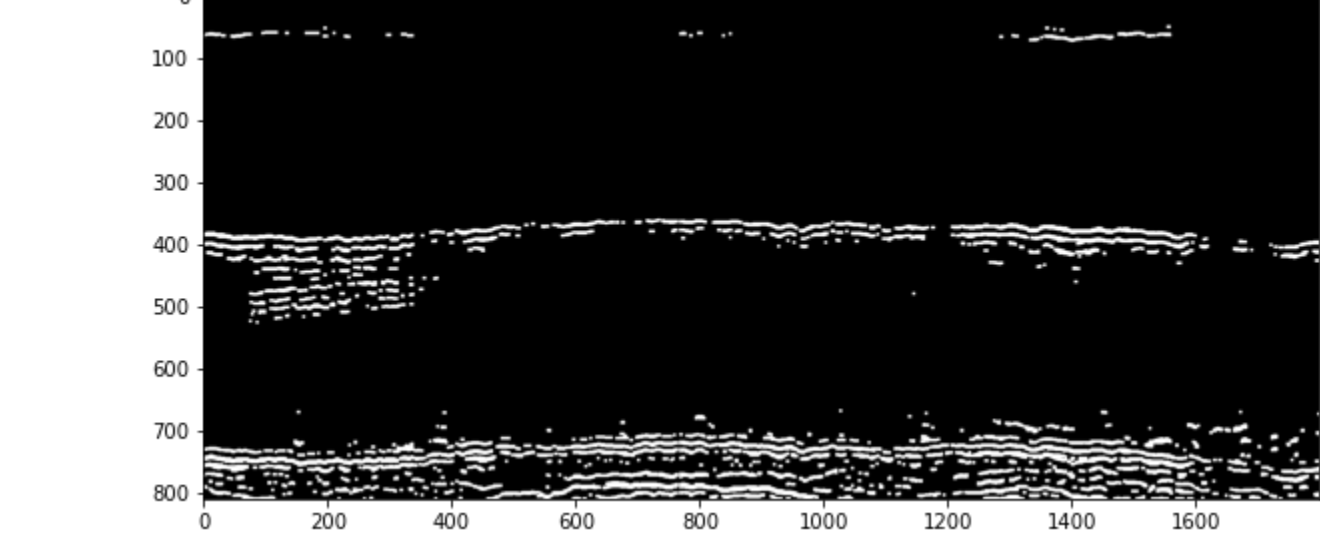
```
In [49]: kernel = np.ones((3,3) , np.uint8)
```

```
In [50]: kernel
```

Out[50]: array([[1, 1, 1],
[1, 1, 1],
[1, 1, 1]], dtype=uint8)

```
In [51]: image_open = cv.morphologyEx(thresh_otsu , cv.MORPH_OPEN , kernel , iterations = 2)
```

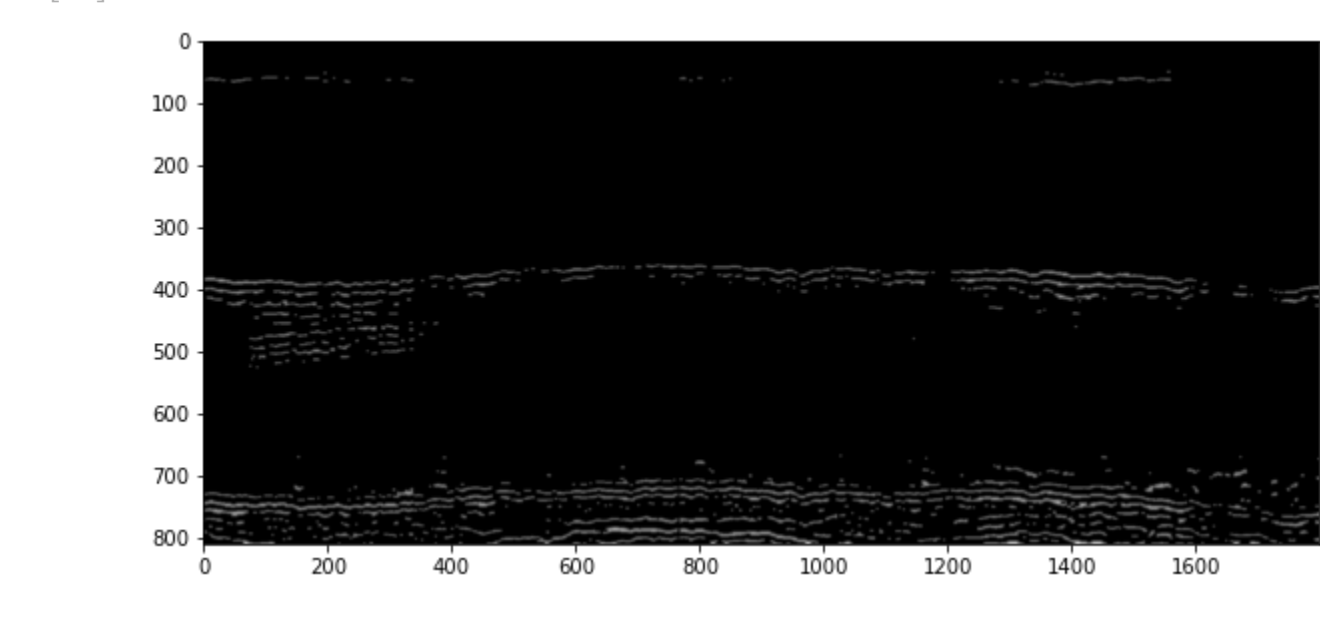
```
In [52]: plt.imshow(image_open , "gray")
```



Applying the distance transform.

```
In [53]: dist_transform = cv.distanceTransform(image_open , cv.DIST_L2,5)
```

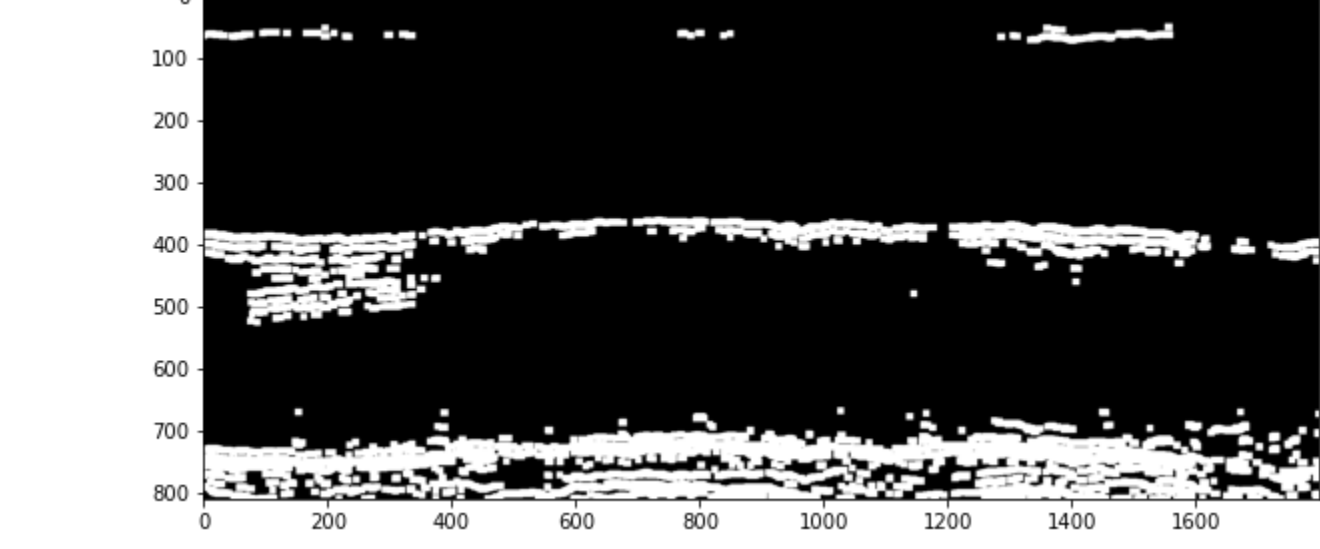
```
In [54]: plt.imshow(dist_transform , "gray")
```



sure background

```
In [55]: sure_bg = cv.dilate(image_open , kernel , iterations = 3)
```

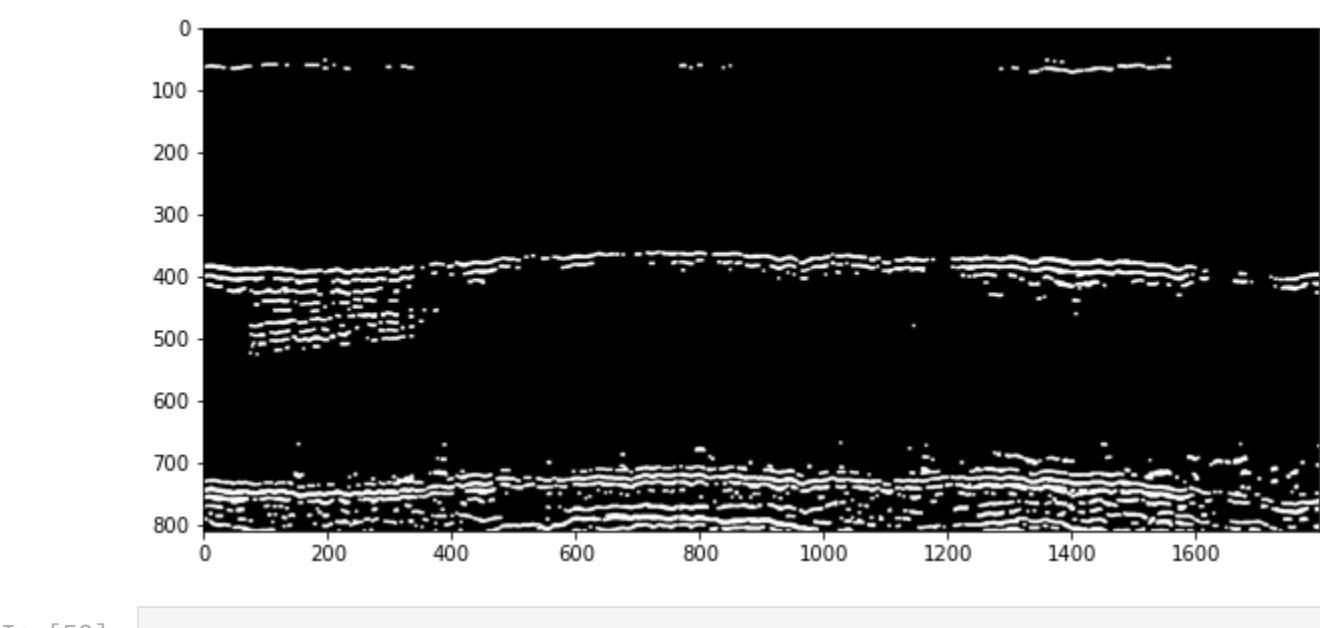
```
In [56]: plt.imshow(sure_bg , "gray")
```



Sure foreground

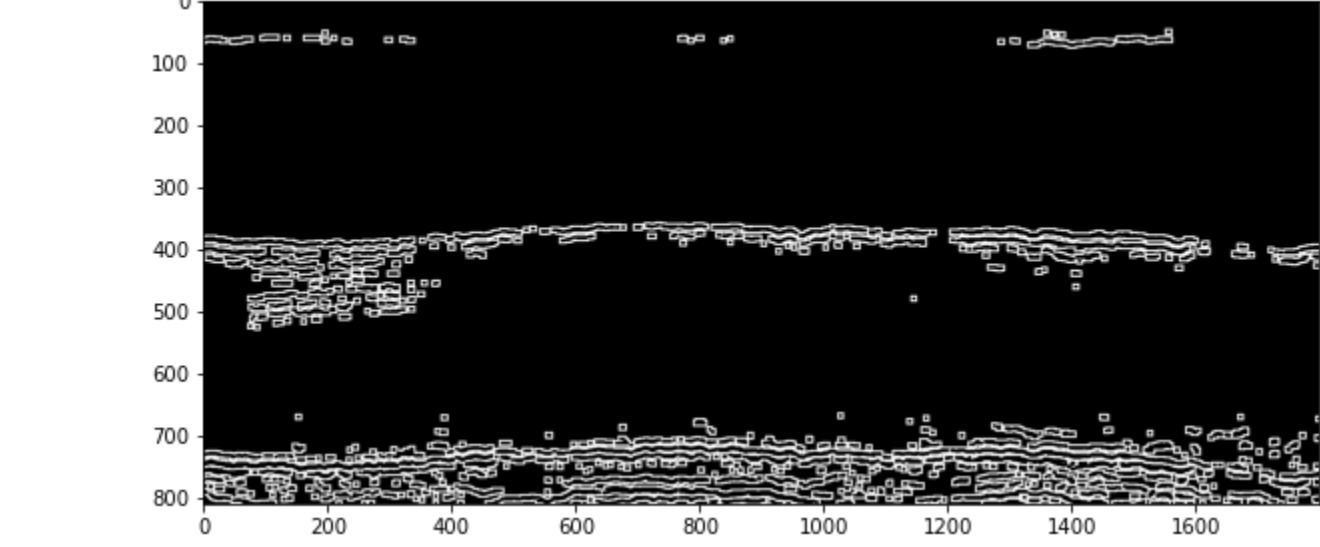
```
In [57]: ret_dis , sure_fg = cv.threshold(dist_transform , 0 , 255 , 0)
```

```
In [58]: plt.imshow(sure_fg,"gray")
```



```
In [59]: unknown = sure_bg - sure_fg
```

```
In [60]: plt.imshow(unknown,"gray")
```



Markers

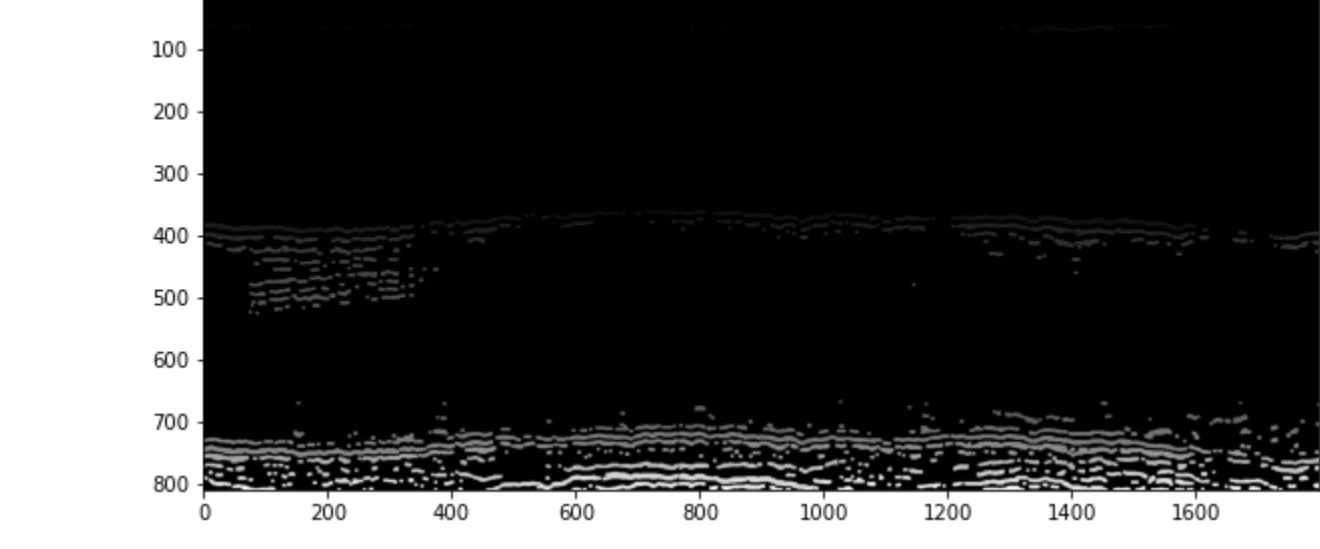
```
In [61]: sure_fg = np.uint8(sure_fg)
```

```
In [62]: ret_mark , markers = cv.connectedComponents(sure_fg)
```

```
In [63]: markers = markers + 1
```

```
In [64]: markers[unknown == 255] = 0
```

```
In [65]: plt.imshow(markers,"gray")
```



```
In [66]: image_watershed = cv.watershed(image , markers)
```

```
error: Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_24176\1851150205.py in <module>
----> 1 image_watershed = cv.watershed(image , markers)

error: OpenCV(4.6.0) D:\a\opencv-python\opencv-python\opencv\modules\imgproc\src\segmentation.cpp:161: error: (-215:Assertion failed) src.type() == CV_8UC3 && dst.type() == CV_32SC1 in function 'cv::watershed'
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```