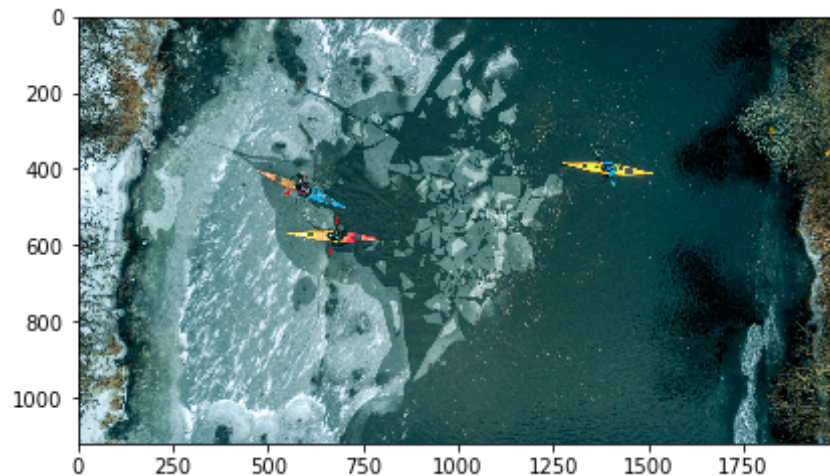


# Image feature extraction

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from skimage.io import imread, imshow
```

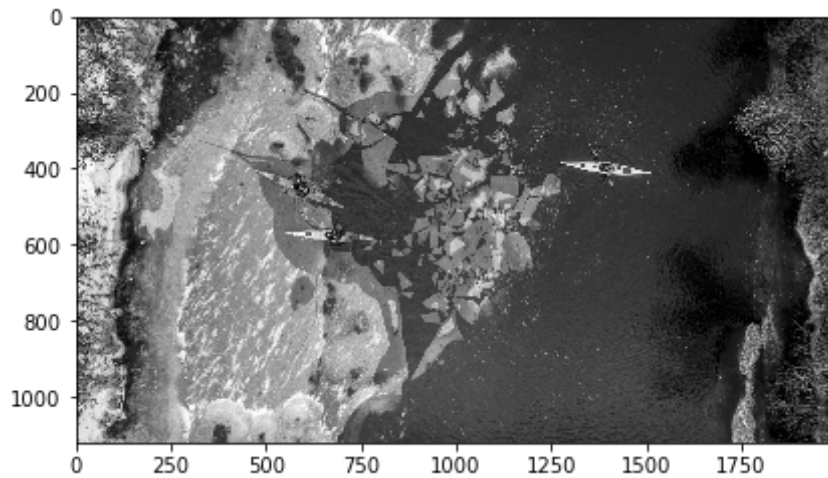
```
In [2]: image1 = imread('image1.jpg')
imshow(image1)
```

Out[2]: <matplotlib.image.AxesImage at 0x1ea89cb0520>



```
In [3]: image2 = imread('image1.jpg', as_gray=True)
imshow(image2)
```

Out[3]: <matplotlib.image.AxesImage at 0x1ea8a3d2d00>



```
In [4]: print(image1.shape)
        print(image2.shape)
```

```
(1121, 1996, 3)
(1121, 1996)
```

```
In [9]: print(image1.size)
        print(image2.size)
```

```
6712548
2237516
```

```
In [14]: pixel_feat1 = np.reshape(image2, (1121 * 1996))
        pixel_feat1
```

```
Out[14]: array([0.24580353, 0.19090157, 0.17521529, ..., 0.2694651 , 0.21456314,
                0.09748157])
```

```
In [15]: pixel_feat2 = np.reshape(image1, (1121 * 1996 * 3))
        pixel_feat2
```

```
Out[15]: array([45, 68, 62, ..., 23, 26, 19], dtype=uint8)
```

## Prewitt

```
In [20]: from skimage.filters import prewitt_h  
from skimage.filters import prewitt_v
```

```
In [21]: pre_hor = prewitt_h(image2)  
pre_ver = prewitt_v(image2)
```

## Sobel

```
In [22]: ed_sobel = filters.sobel(image2)
```

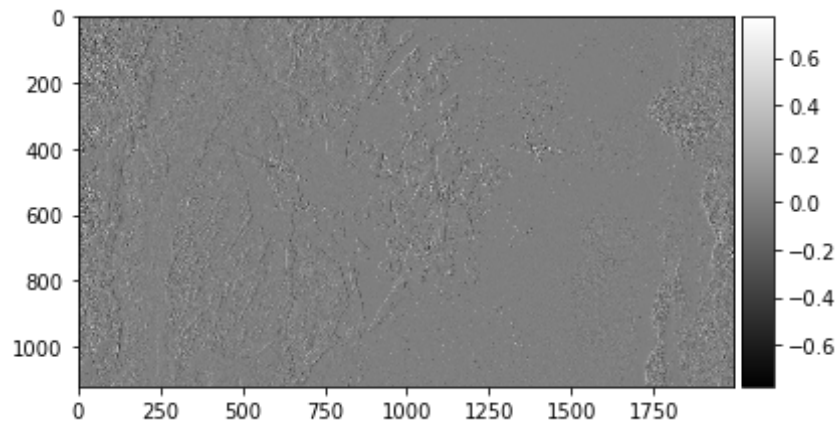
```
In [25]: from skimage import feature
```

## Canny

```
In [26]: can = feature.canny(image2)
```

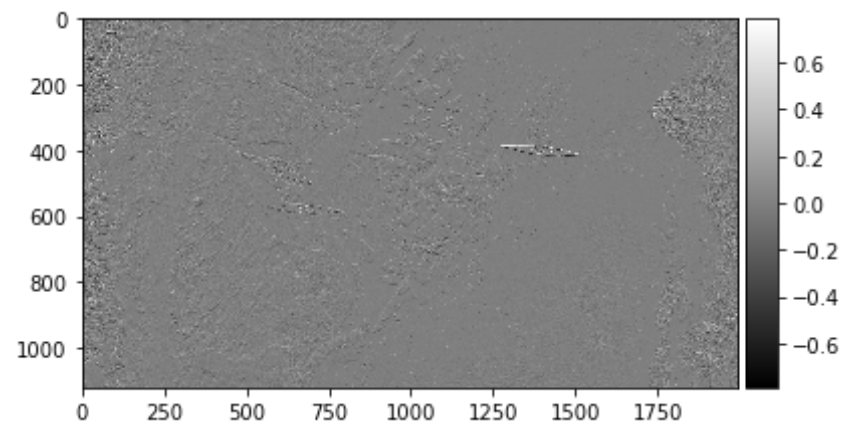
```
In [28]: imshow(pre_ver, cmap='gray')
```

```
Out[28]: <matplotlib.image.AxesImage at 0x1ea8f87af40>
```



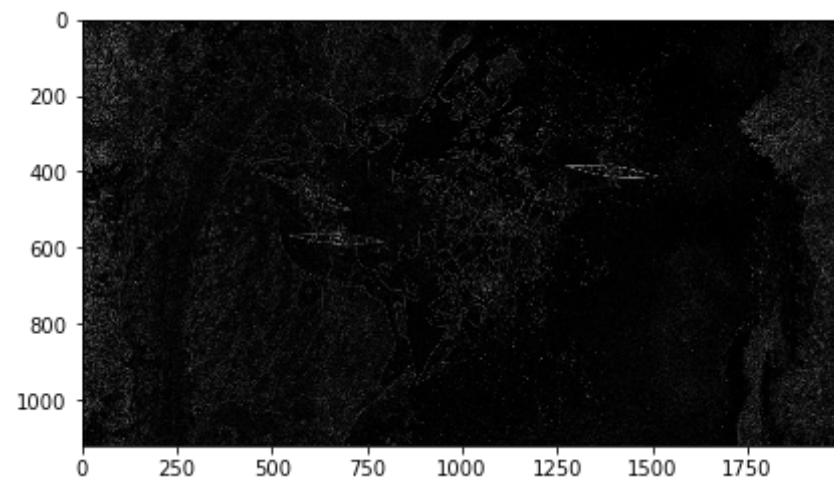
```
In [29]: imshow(pre_hor, cmap='gray')
```

Out[29]: <matplotlib.image.AxesImage at 0x1ea8ffe6eb0>



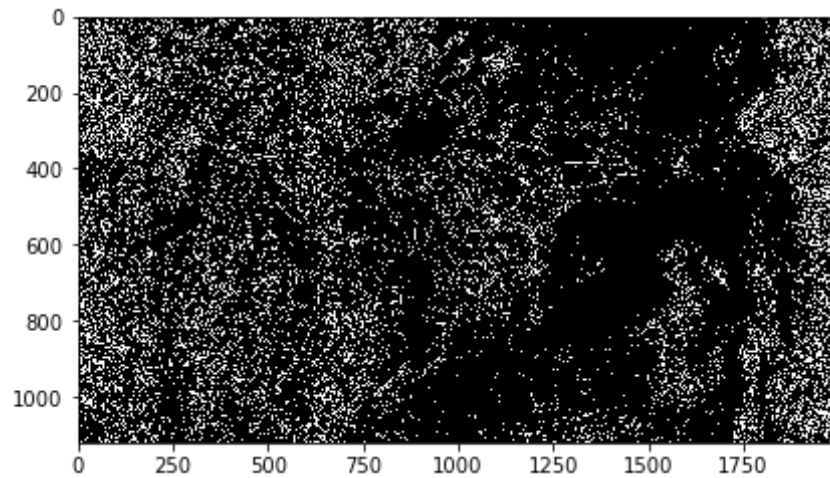
```
In [30]: imshow(ed_sobel, cmap='gray')
```

Out[30]: <matplotlib.image.AxesImage at 0x1ea90079220>



```
In [32]: imshow(can, cmap='gray')
```

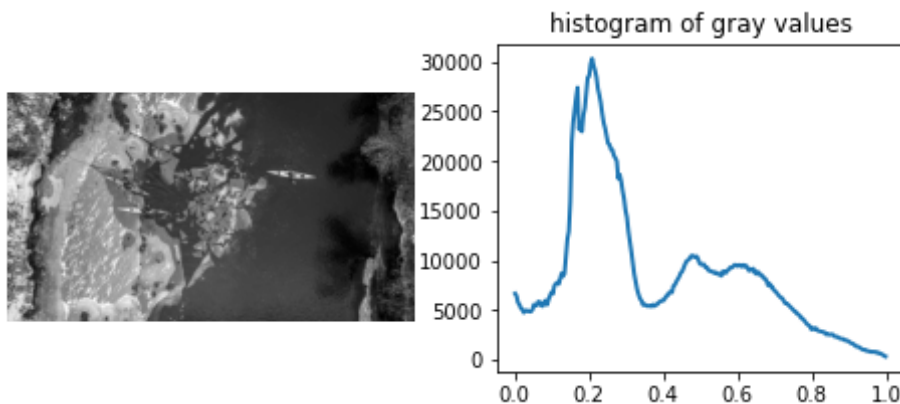
Out[32]: <matplotlib.image.AxesImage at 0x1ea900c54f0>



```
In [33]: from skimage.exposure import histogram
hist, hist_centers = histogram(image2)

#Plotting the Image and the Histogram of gray values
fig, axes = plt.subplots(1, 2, figsize=(8, 3))
axes[0].imshow(image2, cmap=plt.cm.gray)
axes[0].axis('off')
axes[1].plot(hist_centers, hist, lw=2)
axes[1].set_title('histogram of gray values')
```

Out[33]: Text(0.5, 1.0, 'histogram of gray values')



# Flip

```
In [83]: verticalflip = np.flipud(image1)  
         io.imshow(verticalflip)  
         plt.show()
```

