

Звіт
до лабораторної роботи №2
з предмету Комп'ютерне бачення та аналіз зображень

Роботу виконала:

Мерцало Ірина Ігорівна,
студентка групи ПМІМ-11

Відкрила зображення:

```
In [9]: from PIL import Image  
        from PIL import ImageFilter  
        img = Image.open("mountains.jpg")  
        img
```

Out[9]:



Застосувала розмиття по Гаусу, радіус розмиття = 5:

```
In [12]: blur_img = img.filter(ImageFilter.GaussianBlur(5))  
        blur_img
```

Out[12]:



Застосувала медіан-фільтр, який повертає середнє значення з пікселя та його сусідів:

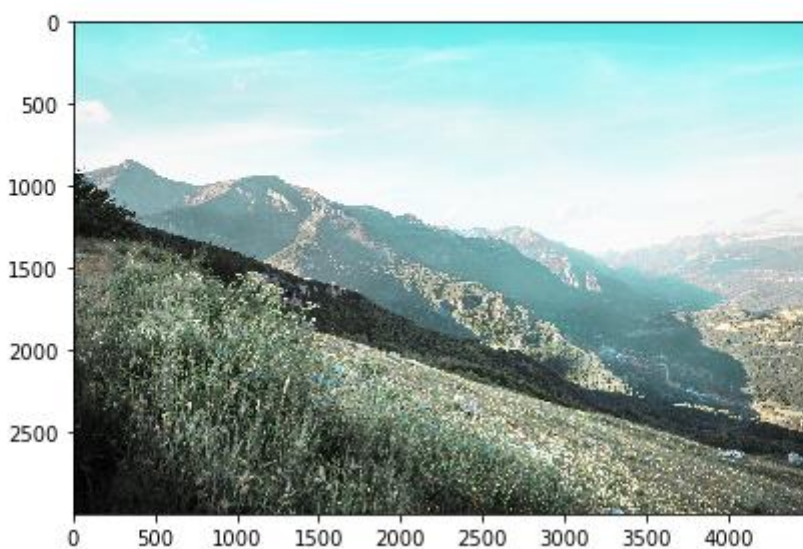
```
In [14]: img = Image.open("mountains.jpg")  
blur_img = img.filter(ImageFilter.MedianFilter(7))  
blur_img
```

Out[14]:



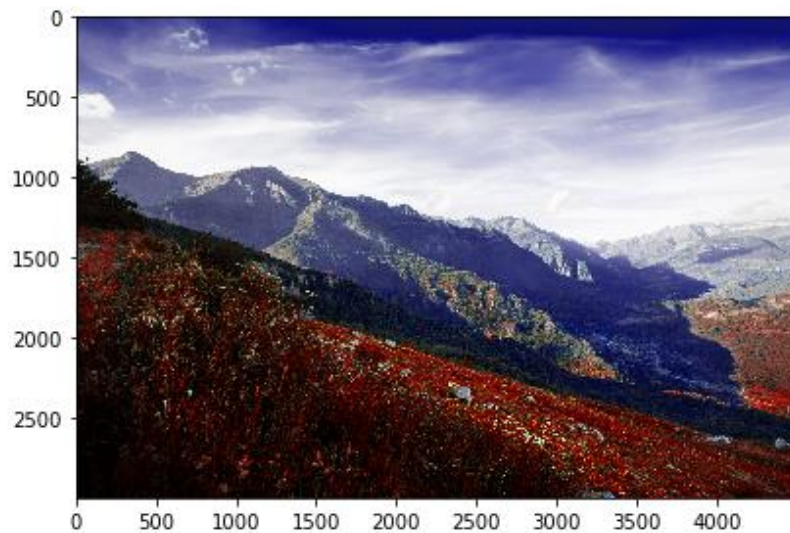
Застосувала такі морфологічні операції, як розширення(щоб заповнити небажані пропуски) та ерозія (щоб забрати шуми, встановивши значення пікселя мінімальним зі значень його сусідів):

```
In [51]: img = io.imread('mountains.jpg')  
dilated_img = morphology.dilation(img)  
io.imshow(dilated_img)  
io.show()
```




```
In [49]: from skimage import io
         from skimage import filters
         from skimage import morphology
         from skimage import data
         from skimage.morphology import disk
         from skimage.filters.rank import median
```

```
In [50]: img = io.imread('mountains.jpg')
         eroded_img = morphology.erosion(img)
         io.imshow(eroded_img)
         io.show()
```



Для застосування власного фільтра, створила ядро і потім задала показники які хотіла б змінити):

```
In [32]: from PIL import ImageFilter
         kernel = ImageFilter.Kernel((3,3), [1,2,3,4,5,6,7,8,9])
```

```
In [35]: from PIL import Image
         from PIL import ImageFilter
         img = Image.open("mountains.jpg")
         img = img.convert("I")
         new_img = img.filter(ImageFilter.Kernel((3,3), [1,0,-1,5,0,-5,1,0,
         new_img
```

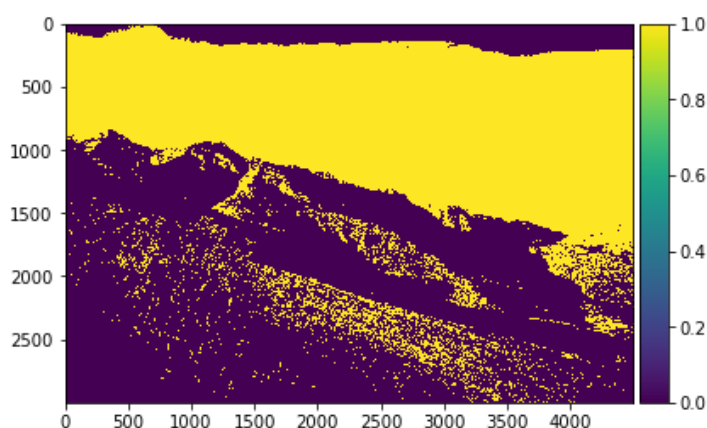
Out[35]:



Корегування зображення відповідно до порогового значення (якщо значення пікселя більше за порогове, то замінити його на піксель білого кольору, якщо менше – то на піксель чорного):

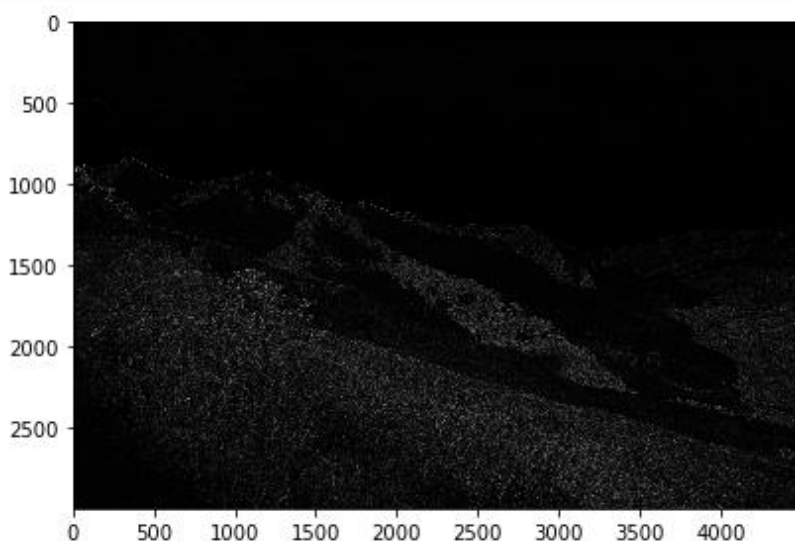
```
In [81]: from skimage.filters import threshold_otsu, threshold_adaptive
from skimage.io import imread, imsave
from skimage.color import rgb2gray

img = imread('mountains.jpg')
img = color.rgb2gray(img)
thresh_value = filters.threshold_otsu(img)
thresh_img = (img > thresh_value).astype(int)
io.imshow(thresh_img)
io.show()
```



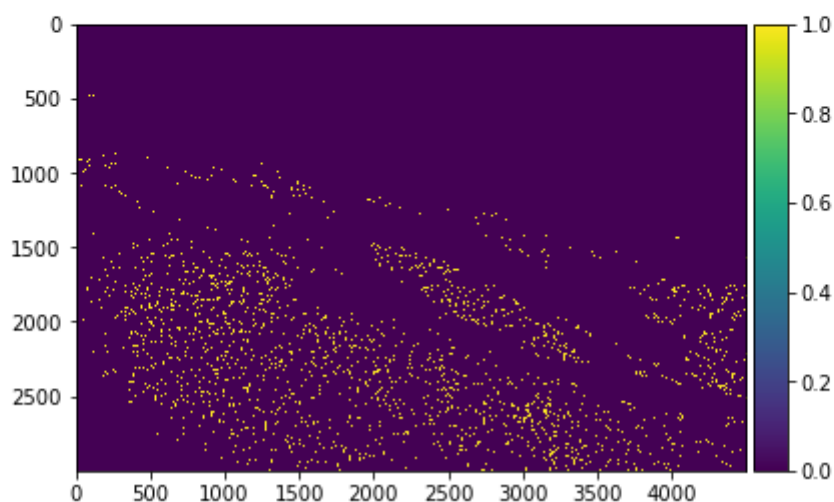
Застосувала виявлення країв (для визначення частин зображення), спершу методом Собеля (визначає магнітуду зміни пікселів):

```
In [38]: from skimage import io
from skimage import filters
from skimage import color
img = io.imread("mountains.jpg")
img = color.rgb2gray(img)
edge = filters.sobel(img)
io.imshow(edge)
io.show()
```



Потім методом Кенні (який використовує не лише магнітуду, а ще напрямок градієнта):

```
In [87]: from skimage import io
from skimage import feature
from skimage import color
img = io.imread("mountains.jpg")
img = color.rgb2gray(img)
edge = feature.canny(img, 3).astype(int)
io.imshow(edge)
io.show()
```



Лінія Хафа:

```
In [93]: from skimage import io
from skimage import transform
image = io.imread('mountains.jpg')
image = color.rgb2gray(image)
probabilistic_lines = transform.probabilistic_hough_line(edge, thre
print(probabilistic_lines)
lines = transform.hough_line(image)
print(lines)
```

```
[((331, 1682), (325, 1679)), ((4149, 2668), (4142, 2666)), ((1
631, 2097), (1618, 2092)), ((1619, 2175), (1610, 2172)), ((144
2, 2266), (1435, 2266)), ((474, 1698), (463, 1694)), ((2781, 2
513), (2776, 2510)), ((1779, 1981), (1773, 1979)), ((3035, 266
4), (3030, 2663)), ((1628, 1929), (1616, 1924)), ((3034, 263
4), (3021, 2628)), ((3493, 2904), (3484, 2900)), ((3799, 276
4), (3794, 2763)), ((962, 1754), (952, 1750)), ((2105, 2447),
(2100, 2446)), ((1284, 2224), (1291, 2224)), ((2372, 2383), (2
360, 2378)), ((2064, 2190), (2058, 2187)), ((4112, 2871), (407
9, 2864)), ((2690, 2579), (2683, 2578)), ((4265, 2151), (4284,
2149)), ((2243, 2476), (2235, 2473)), ((2819, 2298), (2813, 22
96)), ((2464, 2611), (2458, 2608)), ((3717, 2916), (3710, 291
```