

Звіт
до лабораторної роботи №7
з предмету Комп'ютерне бачення та аналіз зображень

Роботу виконала:

Мерцало Ірина Ігорівна,
студентка групи ПМІМ-11

Reading an image:

```
In [8]: import numpy as np
import cv2
from matplotlib import pyplot as plt
%matplotlib inline

img = cv2.imread("mountains.jpg")
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
```



Writing/saving the image:

```
In [9]: img = cv2.imread("mountains.jpg")
cv2.imwrite("mountains_2.jpg", img)
```

Out[9]: True

Результат:



mountains.jpg



mountains_2.jpg

Changing the color space:

```
In [10]: import cv2
img = cv2.imread("mountains.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite("gray_image.jpg", gray)
```

Out[10]: True

Результат:



gray_image.jpg

Scaling:

```
In [11]: import cv2
img = cv2.imread("mountains.jpg")
r,c = img.shape[:2]
new_img = cv2.resize(img, (2*r,2*c), interpolation = cv2.INTER_
cv2.imwrite("resize_image.jpg", new_img)
```

Out[11]: True

Результат:



resize_image.jpg

Cropping the image:

```
In [12]: import cv2
img = cv2.imread("mountains.jpg")
img_crop = img[0:200, 150:350]
cv2.imwrite("crop_img.jpg", img_crop)
```

Out[12]: True

Результат:



crop_img.jpg

Translation:

```
In [13]: import cv2
import numpy as np
img = cv2.imread("mountains.jpg")
r,c = img.shape[:2]
M = np.float32([[1,0,100],[0,1,100]])
new_img = cv2.warpAffine(img,M,(c,r))
cv2.imwrite("translation.jpg", new_img)
```

Out[13]: True

Результат:



translation.jpg

Rotation:

```
In [14]: import cv2
img = cv2.imread("mountains.jpg")
r,c = img.shape[:2]
M = cv2.getRotationMatrix2D((c/2,r/2),90,1)
new_img = cv2.warpAffine(img,M,(c,r))
cv2.imwrite("rotate_img.jpg", new_img)
```

Out[14]: True

Результат:



rotate_img.jpg

Thresholding:

```
In [15]: import cv2
img = cv2.imread("mountains.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
new_img = cv2.threshold(gray,120,255,cv2.THRESH_BINARY)
cv2.imwrite("thresholding.jpg", new_img[1])
```

Out[15]: True

Результат:



thresholding.jpg

Filters:

```
In [16]: import cv2
import numpy as np
img = cv2.imread("mountains.jpg")
ker = np.array([[1, 1, 1],
[1, 1, 1],
[1, 1, 1]])
new_img = cv2.filter2D(img,-1,ker)
cv2.imwrite("filter.jpg", new_img)
```

Out[16]: True

Результат:



filter.jpg

Gaussian blur:

```
In [17]: import cv2  
img = cv2.imread("mountains.jpg")  
new_img = cv2.GaussianBlur(img,(5,5),0)  
cv2.imwrite("gaussian_blur.jpg", new_img)
```

Out[17]: True

Результат:



gaussian_blur.jpg

Median blur:

```
In [19]: import cv2  
img = cv2.imread("mountains.jpg")  
new_img = cv2.medianBlur(img,5)  
cv2.imwrite("median_blur.jpg", new_img)
```

Out[19]: True

Результат:



median_blur.jpg

Morphological operations:

```
In [20]: import cv2
import numpy as np
img = cv2.imread("mountains.jpg")
ker = np.ones((5,5),np.uint8)
new_img = cv2.erode(img,ker,iterations = 1)
cv2.imwrite("erosion.jpg", new_img)
```

Out[20]: True

Результат:



erosion.jpg

Dilation:

```
In [21]: import cv2
import numpy as np
img = cv2.imread("mountains.jpg")
ker = np.ones((5,5),np.uint8)
new_img = cv2.dilate(img,ker,iterations = 1)
cv2.imwrite("dilation.jpg", new_img)
```

Out[21]: True

Результат:



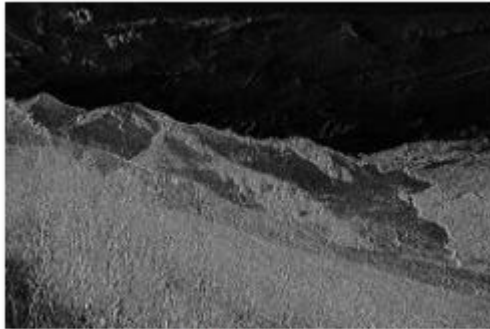
dilation.jpg

Edge detection:

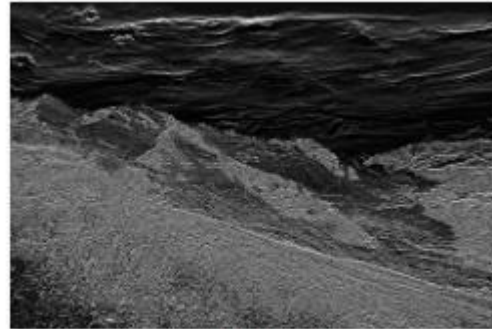

```
In [23]: import cv2
img = cv2.imread("mountains.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
x_edges = cv2.Sobel(gray, -1, 1, 0, ksize=5)
cv2.imwrite("sobel_edges_x.jpg", x_edges)
y_edges = cv2.Sobel(gray, -1, 0, 1, ksize=5)
cv2.imwrite("sobel_edges_y.jpg", y_edges)
```

Out[23]: True

Результат:



sobel_edges_x.jpg



sobel_edges_y.jpg

Canny edge detector:

```
In [24]: import cv2
img = cv2.imread("mountains.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(gray, 100, 200, 3)
cv2.imwrite("canny_edges.jpg", edges)
```

Out[24]: True

Результат:



canny_edges.jpg

Contour detection:


```
In [30]: import cv2
img = cv2.imread("mountains.jpg")
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
thresh_img = cv2.threshold(gray,127,255,0)
im, contours = cv2.findContours(thresh_img[1],cv2.RETR_TREE,cv2
cv2.drawContours(img, im, -1, (255,0,0), 3)
cv2.imwrite("contours.jpg", img)
```

Out[30]: True

Результат:



contours.jpg

Template matching:

```
In [32]: import cv2
img = cv2.imread("mountains.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img_temp = cv2.imread("crop_img.jpg")
gray_temp = cv2.cvtColor(img_temp, cv2.COLOR_BGR2GRAY)
w, h = gray_temp.shape[: -1]
output = cv2.matchTemplate(gray,gray_temp,cv2.TM_CCOEFF_NORMED)
min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(output)
top = max_loc
bottom = (top[0] + w, top[1] + h)
cv2.rectangle(img,top, bottom, 255, 2)
cv2.imwrite("img.jpg",img)
```

Out[32]: True

Результат:



crop_img.jpg



img.jpg