

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Морфологические преобразования»

ОТЧЕТ
по лабораторной работе №14
дисциплины
«Технологии распознавания образов»

Выполнила:

Кувшин Ирина Анатольевна
2 курс, группа ПИЖ-б-о-21-1,
010.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Цель работы:

изучение различных морфологических операций, таких как эрозия, расширение, открытие, закрытие и т. д. Приобретение навыков работы с функциями: `cv2.erode ()`, `cv2.dilate ()`, `cv2.morphologyEx ()`.

Ход выполнения:


IRT_14 - Jupyter_Lab14.ipynb

Project: IRT_14, Jupyter_Lab14.ipynb, Jupyter_Lab14.ipynb, Jupyter_Lab14.ipynb, README.md, requirements.txt, External Libraries, Scratches and Consoles

8.1. Дилатация (расширение) и 8.2. Эрозия

```
In 62 1 image = img_input('img/sun.png', type=0)
2
3 # Создадим ядро 5x5
4 kernel = np.ones((5, 5), np.uint8)
5
6 # Выполним функции дилатации и эрозии
7 dilation = cv2.dilate(image, kernel, iterations=2)
8 erosion = cv2.erode(image, kernel, iterations=2)
9
10 res = [image, dilation, erosion]
11 title = ['Оригинал', 'Дилатация', 'Эрозия']
12
13 img_print(res, title, hard_type='gray')
```

Оригинал Дилатация Эрозия



IRT_14 - Jupyter_Lab14.ipynb

Project: IRT_14, Jupyter_Lab14.ipynb, Jupyter_Lab14.ipynb, README.md, requirements.txt, External Libraries, Scratches and Consoles

8.3. Открытие, размыкание и 8.4. Закрытие, замыкание

```
In 63 1 image = Image.open('img/sun.jpg')
2 draw = ImageDraw.Draw(image)
3 width = image.size[0]
4 height = image.size[1]
5 pix = image.load()
6 for i in range(width):
7     for j in range(height):
8         rand = random.randint(0, 150)
9         a = pix[i, j][0] + rand
10        b = pix[i, j][1] + rand
11        c = pix[i, j][2] + rand
12        if (a > 255):
13            a = 255
14        if (b > 255):
15            b = 255
16        if (c > 255):
17            c = 255
18        draw.point((i, j), (a, b, c))
19
20 image.save('img/median.png', "JPEG")
21
22 noise = cv2.imread("img/median.png", 1)
23
24 kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (30, 30))
```

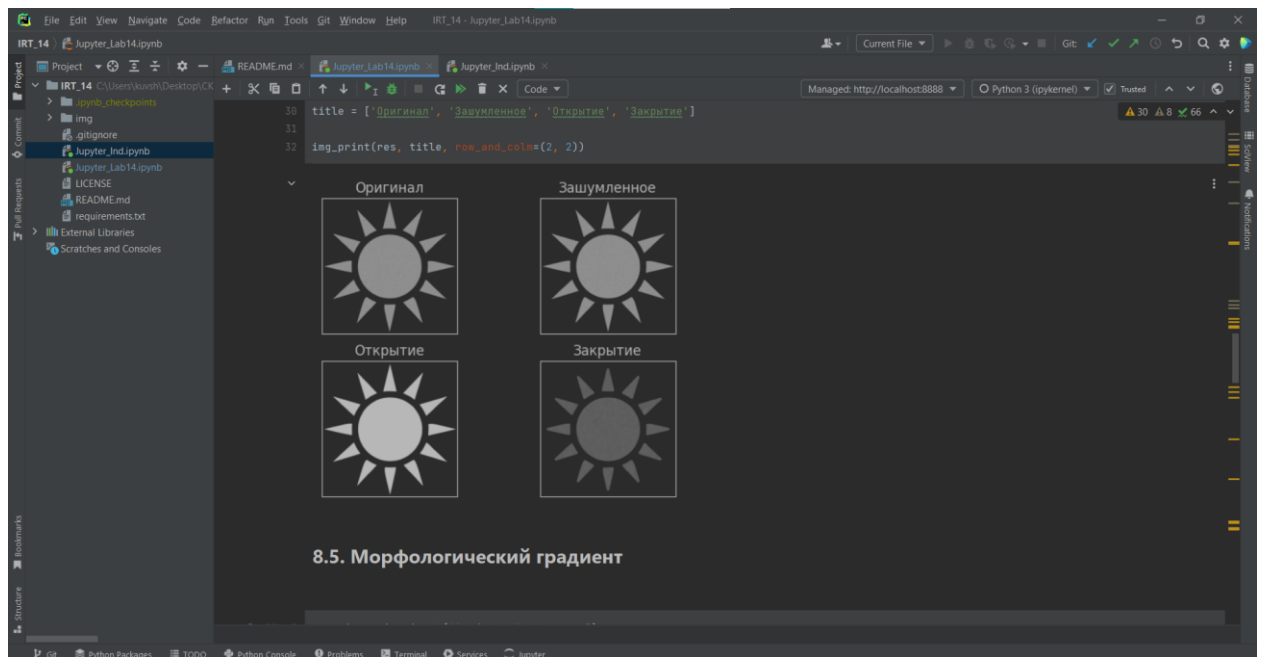


Рисунок 14.1– Код программы

Индивидуальное задание

8. Морфологические преобразования - Individual

```

In 20 1 import ...

In 21 1 # функция ввода изображения
      2 def img_input(img_path, size=(0, 0), type=1):...
      17
      18
      19 # функция вывода изображения
      20 def img_print(images, titles, row_and_col=(0, 0), hard_type='None'):...

```

1. Загрузить библиотеку питру, файл bin.jpg и преобразовать его с помощью операций дилатация и эрозия. Выбрать ядро, размер которого равен последней цифре в номере списка группы.

```

In 22 1 image = img_input('img/ind_1.jpg', type=0)
      2
      3 # Создадим ядро 5x5
      4 kernel = np.ones((5, 5), np.uint8)
      5
      6 # Выполним функции дилатации и эрозии

```

Оригинал

Дилатация

Эрозия

2. Для демонстрации удаления шума создать зашумленный файл, затем к зашумленному файлу применить операцию открытия и закрытия.

```

In 23 1 image = Image.open('img/ind_cat.jpg')
      2 draw = ImageDraw.Draw(image)
      3 width = image.size[0]
      4 height = image.size[1]
      5 pix = image.load()
      6 for i in range(width):
      7     for j in range(height):
      8         rand = random.randint(0, 150)
      9         a = pix[i, j][0] + rand
      10        b = pix[i, j][1] + rand
      11        c = pix[i, j][2] + rand
      12        if (a > 255):
      13            a = 255
      14        if (b > 255):
      15            b = 255

```

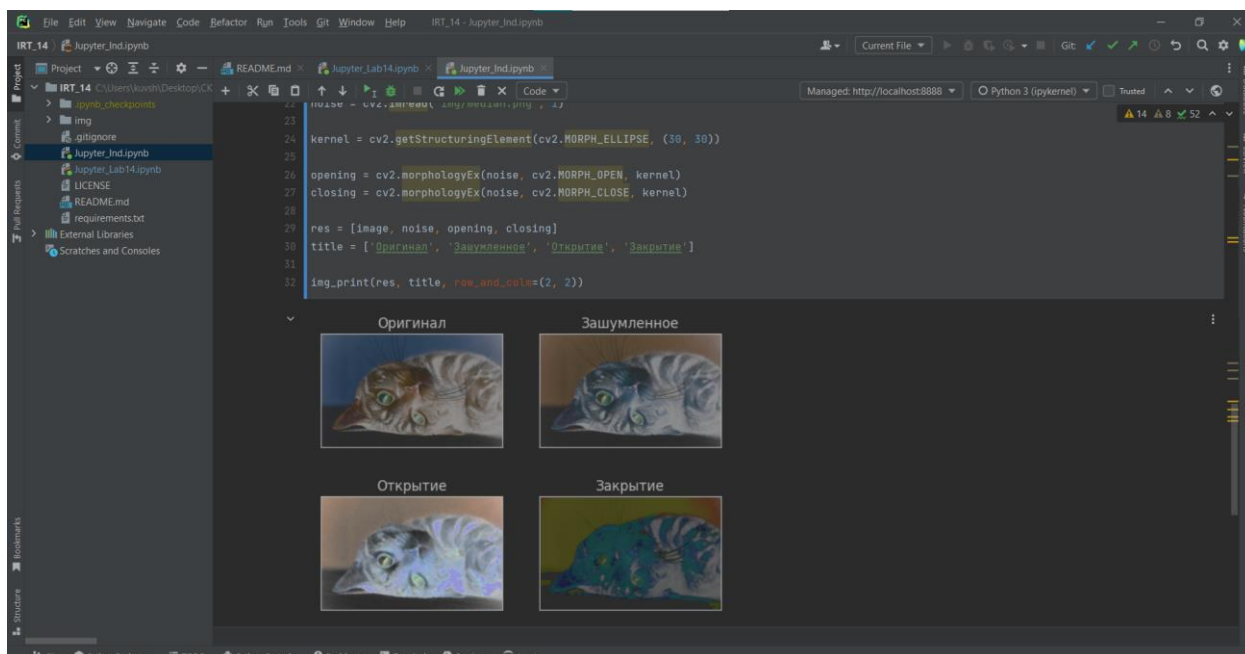


Рисунок 14.2– Код программы

Контрольные вопросы:

1. Что такое морфологические преобразования?

Морфологические преобразования – это преобразования множества пикселей одного изображения с помощью множества пикселей другого – структурирующего элемента или ядра.

2. Сколько существует основных морфологических операторов?

Имеются два основных морфологических оператора – это эрозия и расширение. Применение этих операторов в различных комбинациях дает новые морфологические преобразования: открытие, закрытие и т. Д

3. Какие аргументы принимает функция дилатации?

Первый аргумент функции дилатации – это массив основного множества, второй – это ядро:

dilation = cv2.dilate(img, kernel, iterations = 1).

4. Что такое операция Эрозия?

Операция эрозия – это объединение всех центральных пикселей ядра В, которое не выходит за пределы основного множества А.

5. Что делает операция эрозии с изображением?

Операция эрозия ужимает (уточняет) изображения.

6. Какие аргументы принимает функция эрозии?

Первый аргумент функции эрозии – это массив основного множества, второй – это ядро:

```
erosion = cv2.erode(img,kernel,iterations = 1).
```

7. Что такое операция открытия?

Операция открытия, размыкания – это объединение всех множеств примитивов, находящихся внутри основного множества.

8. Какая функция используется для открытия?

Для открытия используется функция `cv2.morphologyEx ()`
`opening=cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel).`

9. Для чего полезна операция открытия?

Операция открытия полезна для удаления шума.

10. В чем отличие замыкания от размыкания?

Замыкание также сглаживает участки контура, но в отличие от размыкания «заливает» узкие разрывы и углубления малой ширины, ликвидирует небольшие отверстия или маленькие черные точки на объекте.