МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 3.14

«Морфологические преобразования»

по дисциплине «Технологии распознавания образов»

Выполнил студент группы
ПИЖ-б-о-21-1
Зиберов Александр
« » мая 2023 г.
Подпись студента_____
Работа защищена
« »_____20__г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь, 2023

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().

Выполнение работы:

Проработать примеры лабораторной работы в отдельном ноутбуке.

Задание 8.1.

Загрузить библиотеку numpy, файл bin.jpg и преобразовать его с помощью операций дилатация и эрозия. Выбрать ядро, размер которого равен последней цифре в номере списка группы. Здесь ядро 5×5. Выполним сначала операцию дилатации, затем и эрозии

In [1]:
```
import cv2
import numpy as np
import random
from PIL import Image, ImageDraw
from matplotlib import pyplot as plt
```

In [2]:
```
img = cv2.imread('pictures/sun.png',0)

kernel = np.ones((5,5), np.uint8)

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

plt.figure(figsize=(20,15))

plt.subplot(131),
plt.imshow(img,cmap = 'gray'),plt.title("Оригинал"),
plt.axis('off')

plt.subplot(132),
plt.imshow(dilation,cmap = 'gray'),plt.title("Дилатация"),
plt.axis('off')

plt.subplot(133),
plt.imshow(erosion,cmap = 'gray'),plt.title("Эрозия"),
plt.axis('off');
```

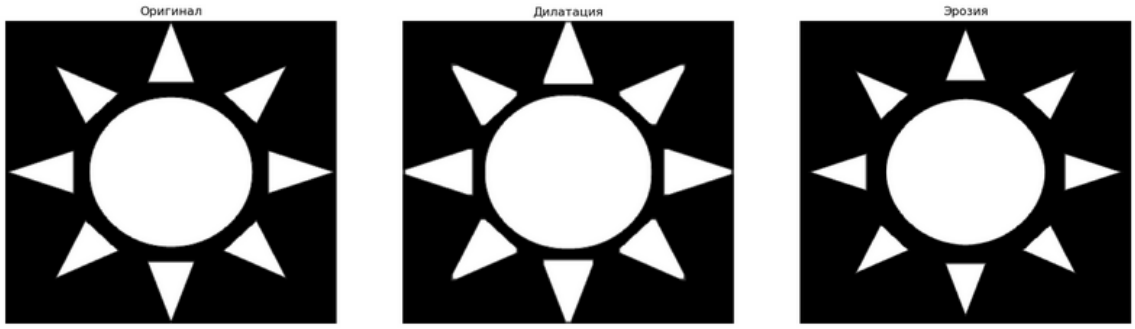


Рисунок 1 – Пример 1

Задание 8.2.

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

In [3]:
```
image = Image.open('pictures/star.jpg')

draw = ImageDraw.Draw(image)
width = image.size[0]
height = image.size[1]
pix = image.load()
for i in range(width):
    for j in range(height):
        rand = random.randint(0, 150)
        a = pix[i, j][0] + rand
        b = pix[i, j][1] + rand
        c = pix[i, j][2] + rand
        if (a > 255):
            a = 255
        if (b > 255):
            b = 255
        if (c > 255):
            c = 255
        draw.point((i, j), (a, b, c))
image.save("pictures/median.png", "JPEG")

image = Image.open('pictures/star.jpg')

median = cv2.imread("pictures/median.png", 1)
median = cv2.cvtColor(median,cv2.COLOR_BGR2RGB)

kernel = cv2.getStructuringElement(cv2.MORPH_RECT,(10, 10))
opening = cv2.morphologyEx(median, cv2.MORPH_OPEN, kernel)

plt.figure(figsize=(15,15))

plt.subplot(131),
plt.imshow(image,cmap = 'gray'),plt.title("Оригинал"),
plt.axis('off')

plt.subplot(132)
plt.imshow(median, cmap='gray'),plt.title("Шум")
plt.axis('off')

plt.subplot(133)
plt.imshow(opening, cmap='gray'),plt.title("Открытие")
plt.axis('off');
```

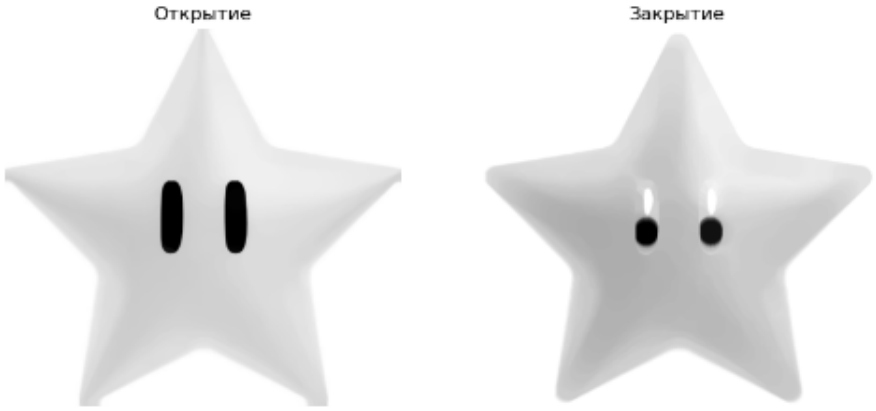
Оригинал Шум Открытие

Рисунок 2 – Пример 2

Задание 8.3.

Трансформировать цветное изображение в полутоновое при его загрузке, к полутоновому файлу применить операцию открытия.

In [4]:
```
img = cv2.imread('pictures/star_n.jpg',0)

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(30,30))

opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
close = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)

plt.figure(figsize=(10,10))

plt.subplot(121)
plt.imshow(opening, cmap='gray')
plt.title("Открытие")
plt.axis('off')

plt.subplot(122)
plt.imshow(close, cmap='gray')
plt.title("Закрытие")
plt.axis('off')

plt.show();
```



Задание 8.4.

Трансформировать цветное изображение в полутоновое при его загрузке. Скопировать полутоновое изображение. К первому изображению применить операцию расширения, ко второму эрозию. Затем вычесть из расширенного изображения изображение после эрозии. Результат похож на контур объекта.

In [5]:
```
img = cv2.imread('pictures/fox_2.jpg',0)

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img1 = img.copy()

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5,5))

dilation = cv2.dilate(img, kernel, iterations = 1)
erosion = cv2.erode(img1, kernel, iterations = 1)

result = dilation - erosion

plt.figure(figsize=(15,10))

plt.subplot(221)
plt.imshow(img1, cmap='gray')
plt.title("Оригинал")
plt.axis('off')

plt.subplot(222)
plt.imshow(result, cmap='gray')
plt.title("Расширение")
plt.axis('off')

plt.show()
```

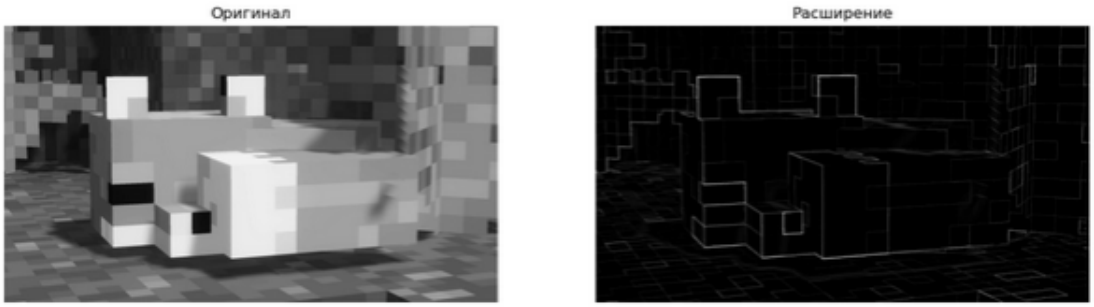
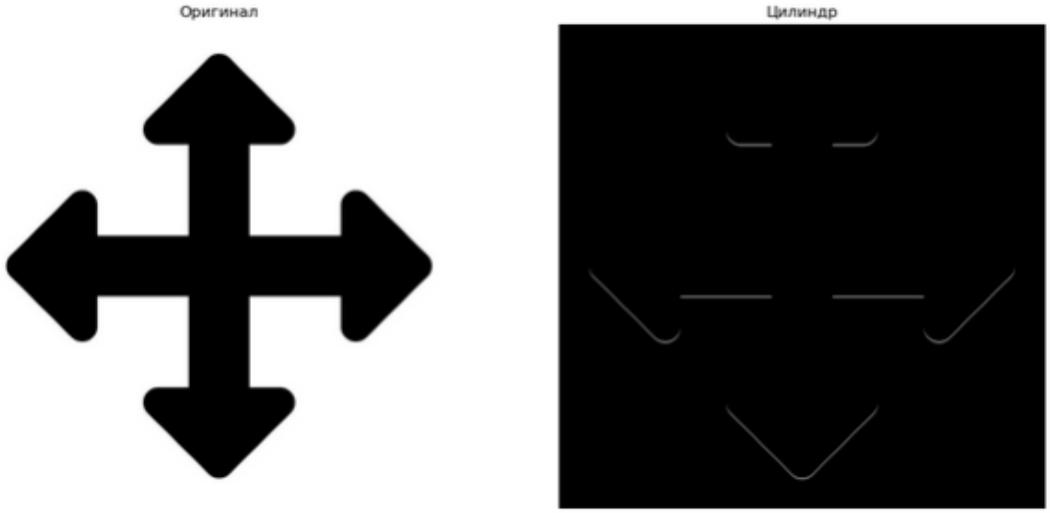


Рисунок 3 – Пример 3

Задание 8.5.

Применить операцию цилиндр к изображению, размер ядра равен 40 + №, № – номер по списку группы. (№7, 40 + 7 = 47)

In [6]:
```
img = cv2.imread('pictures/cross.jpg')
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)

result = cv2.morphologyEx(img, cv2.MORPH_TOPHAT, (47,47))

plt.figure(figsize=(15,10))
plt.subplot(121)
plt.imshow(img, cmap="gray")
plt.title("Оригинал")
plt.axis('off')

plt.subplot(122)
plt.imshow(result, cmap="gray")
plt.title("Цилиндр")
plt.axis('off')

plt.show();
```



Задание 8.6.

Применить операцию черная шляпа к изображению, размер ядра равен 40 + №, № – номер по списку группы. (№7, 40 + 7 = 47)

In [7]:
```
result = cv2.morphologyEx(img, cv2.MORPH_BLACKHAT, (47,47))

plt.figure(figsize=(15,10))
plt.subplot(121)
plt.imshow(img, cmap="gray")
plt.title("Оригинал")
plt.axis('off')

plt.subplot(122)
plt.imshow(result, cmap="gray")
plt.title("Черная шляпа")
plt.axis('off')

plt.show();
```

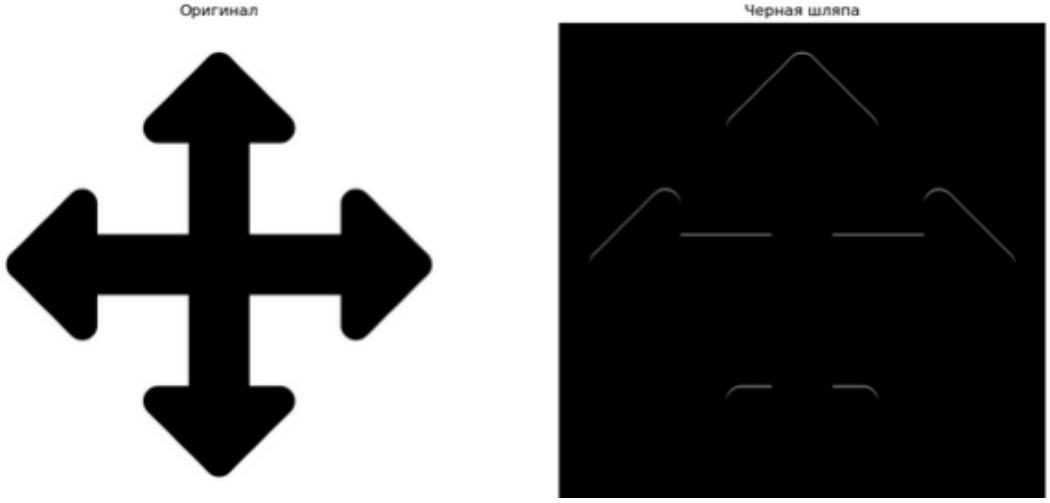


Рисунок 4 – Пример 4

Задание 8.7.

Изготовить ядро, его размер выбрать из ряда 3/3, 3/5, 5/3, 5/5, 5/7, 3/7, 7/3, 7/5, 5/7, 7/7, номер варианта должен быть равен номеру по списку группы. Обработать изображение с помощью выбранного ядра и ядра размером 9/9. Сравнить результаты обработки изображения этими ядрами.
Вариант 7 = 7/3

In [8]:
```
img = cv2.imread('pictures/nah.jpg',0)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (7, 3))
result = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)

kernel1 = cv2.getStructuringElement(cv2.MORPH_RECT, (9, 9))
result1 = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel1)

plt.figure(figsize=(15,10))
plt.subplot(121), plt.imshow(result), plt.title('7x5')
plt.axis('off')
plt.subplot(122), plt.imshow(result1), plt.title('9x9')
plt.axis('off')
plt.show();
```

Рисунок 5 – Пример 5

Индивидуальное задание

Индивидуальное задание

1. Увеличить толщину текста на изображении морфологическими преобразованиями
2. Улучшить видимость капчи, убрав шум морфологическими преобразованиями

In [241]:
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

def imgshow(image, conversion=cv2.COLOR_BGR2RGB):
    image = cv2.cvtColor(image, conversion)
    plt.imshow(image)
    plt.axis("off")
    plt.xticks([])
    plt.yticks([])
    plt.show()
```

Задание 1.

Операция эрозия позволяет увеличить темные области изображения, что подойдет для изменения толщины текста.
Для этого сначала создадим ядро (kernel), а после применим функцию cv2.erode() к изображению.

In [242]:
```
img = cv2.imread('1_b.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

kernel = np.ones((5,5), np.uint8)

erosion = cv2.erode(img, kernel,iterations = 3)

plt.figure(figsize=(20,15))

plt.subplot(131),
plt.imshow(img,cmap = 'gray'),plt.title("Оригинал"),
plt.axis('off')

plt.subplot(132),
plt.imshow(erosion,cmap = 'gray'),plt.title("Эрозия"),
plt.axis('off');
```

Рисунок 6 – Индивидуальное задание (1)

In [243]:
```
img2 = cv2.imread('1_something.jpg')
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)

erosion2 = cv2.erode(img2, kernel,iterations = 1)

plt.figure(figsize=(20,15))

plt.subplot(131),
plt.imshow(img2,cmap = 'gray'),plt.title("Оригинал"),
plt.axis('off');

plt.subplot(132),
plt.imshow(erosion2,cmap = 'gray'),plt.title("Эрозия"),
plt.axis('off');
```

Задание 2.

Метод открытие (комбинация операций эрозии и расширения) может помочь в удалении шума.
Применим его к капче функцией cv2.morphologyEx() с параметром cv2.MORPH_CLOSE.

In [244]:
```
img = cv2.imread('captcha.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

kernel = np.ones((5,5),np.uint8)

close = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)

plt.figure(figsize=(5,5))

plt.subplot(121)
plt.imshow(img)
plt.title("Оригинал")
plt.axis('off')

plt.subplot(122)
plt.imshow(close)
plt.title("Закрытие")
plt.axis('off')

plt.show();
```

Увеличим видимость, изменив гамму изображения

In [245]:
```
def make_darker(img, gamma):
    gamma_table=[np.power(x/255.0,gamma)*255.0 for x in range(256)]
    gamma_table=np.round(np.array(gamma_table)).astype(np.uint8)
    return cv2.LUT(img,gamma_table)

close = make_darker(close, 2)

plt.figure(figsize=(3,3))
plt.imshow(close)
plt.axis('off')
plt.show();
```

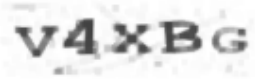


Рисунок 7 – Индивидуальное задание (2)

Вывод: В результате выполнения работы были изучены морфологических преобразований.

1. Что делает операция дилатации?

Увеличивает размер объекта на изображении

2. Что делает операция эрозии?

Уменьшает размер объекта на изображении.

3. Что делает операция градиента?

Разницу между дилатацией и эрозией, используется для выделения границ объектов

4. Что делает операция TOP_HAT?

Разница между исходным изображением и открытием, используется для выделения мелких объектов на фоне

5. Что делает операция BLACK_HAT?

Разница между закрытием и исходным изображением, так же используется для выделения мелких объектов на фоне

6. Что делает операция открытия?

Сочетание эрозии и дилатации, используется для удаления мелких объектов и зашумления.

7. Что делает операция закрытия?

Сочетание дилатации и эрозии, используется для заполнения небольших полостей в объектах и зашумления

8. Какого рода шумы лучше всего устраняют операции открытия/закрытия?

Соль/Перец.

9. Что делает функция cv2.getStructuringElement()?

Создаёт ядро(матрицу), заданной размерности, с учётом указанной формы.

10. Что такое Морфологические преобразования?

Операций, основанные на форме изображения. Обычно подобные преобразования выполняются над двоичными изображениями. Ему нужны два входа, один — это исходное изображение, второй называется структурным элементом или ядром, которое определяет характер операции.