

**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
по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1

Трушева В. О. .«__»_____2023г.

Подпись студента_____

Работа защищена « __ »_____20__г.

Проверила Воронкин Р.А. _____
(подпись)

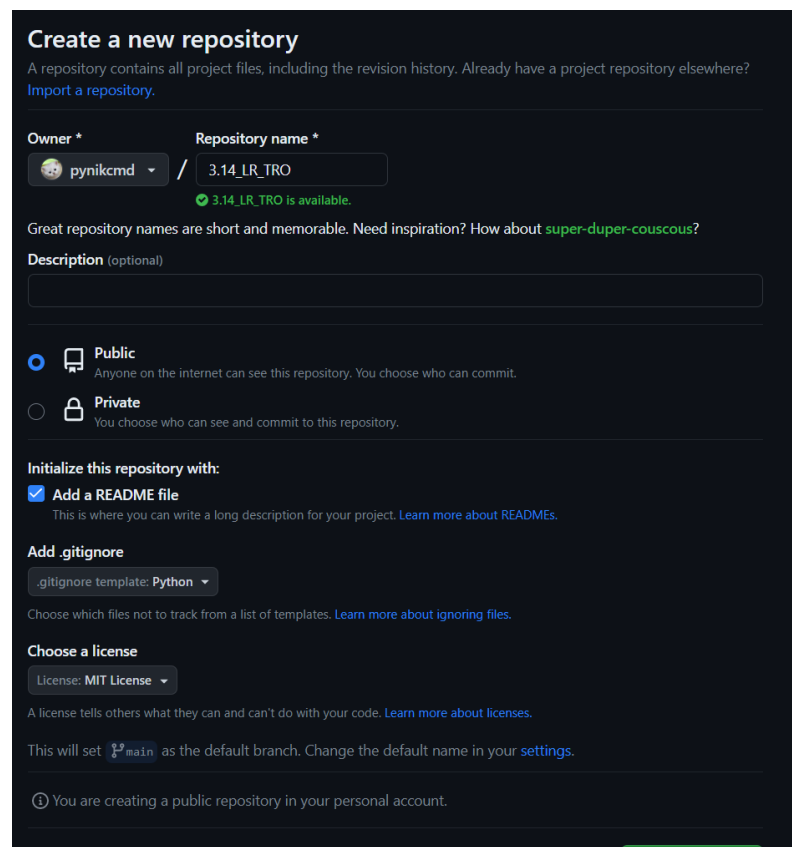
Ставрополь 2023

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()`.

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.

2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).



The screenshot shows the GitHub 'Create a new repository' page. The 'Owner' is set to 'pynikcmd' and the 'Repository name' is '3.14_LR_TRO'. A green checkmark indicates that the name is available. The 'Description' field is empty. The 'Public' option is selected under 'Visibility'. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked. The '.gitignore' template is set to 'Python'. The 'License' is set to 'MIT License'. At the bottom, a note states: 'You are creating a public repository in your personal account.'

3. Выполните клонирование созданного репозитория на рабочий компьютер.

```
D:\fgit>git clone https://github.com/pynikcmd/3.14_LR_TRO.git
Cloning into '3.14_LR_TRO'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
2Receiving objects: 100% (5/5), done.
```

4. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
D:\fgit\3.14_LR_TRO>git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [D:/fgit/3.14_LR_TRO/.git/hooks]
```

5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

6. 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.

```
import cv2
import numpy as np
import random
from PIL import Image, ImageDraw
from matplotlib import pyplot as plt
```

```
img = cv2.imread('img/img.png',0)
```

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

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

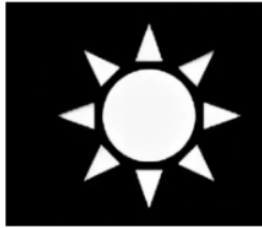
```
plt.figure(figsize=(10,10))

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

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

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

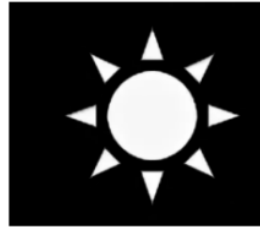
Original



Dilation



Erosion



Задание 8.2

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

```
: image = Image.open('img/img.png')

: 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("img/median.png", "PNG")
median = cv2.imread("img/median.png", 1)

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

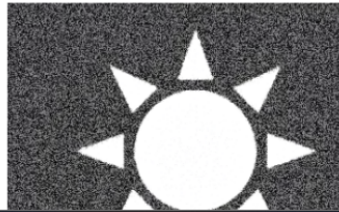
plt.subplot(122),
plt.imshow(median,cmap = 'gray'),plt.title("Modified"),
plt.axis('off')

plt.show();
```

Original



Modified



Задание 8.3

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

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

```
] open = cv2.morphologyEx(median, cv2.MORPH_OPEN, kernel)  
clos = cv2.morphologyEx(median, cv2.MORPH_CLOSE, kernel)
```

```
] plt.figure(figsize=(10,10))  
  
plt.subplot(121)  
plt.imshow(open, cmap='gray')  
plt.title("Opening")  
plt.axis('off')  
  
plt.subplot(122)  
plt.imshow(clos, cmap='gray')  
plt.title("Closing")  
plt.axis('off')  
  
plt.show();
```

Opening



Closing



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.

```
image = cv2.imread('img/img2.jpg')

gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

copied_image = gray_image.copy()

kernel = np.ones((5, 5), np.uint8)
dilated_image = cv2.dilate(gray_image, kernel, iterations=1)

eroded_image = cv2.erode(copied_image, kernel, iterations=1)

result_image = dilated_image - eroded_image

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

plt.subplot(331)
plt.imshow(image, cmap='gray')
plt.title("Original")
plt.axis('off')

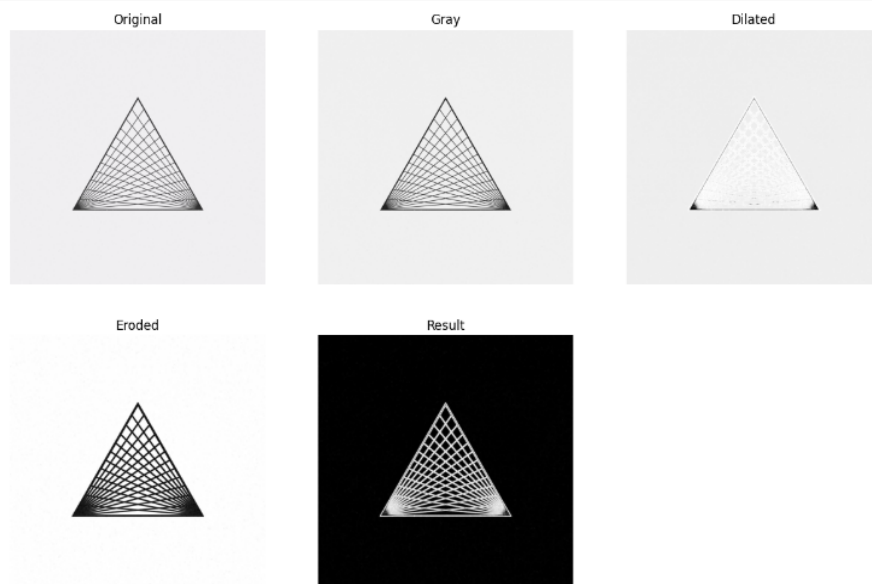
plt.subplot(332)
plt.imshow(gray_image, cmap='gray')
plt.title("Gray")
plt.axis('off')

plt.subplot(333)
plt.imshow(dilated_image, cmap='gray')
plt.title("Dilated")
plt.axis('off')

plt.subplot(334)
plt.imshow(eroded_image, cmap='gray')
plt.title("Eroded")
plt.axis('off')

plt.subplot(335)
plt.imshow(result_image, cmap='gray')
plt.title("Result")
plt.axis('off')

plt.show()
```



Задание 8.5.

Применить операцию цилиндр к изображению, размер ядра равен $40 + N_0$, N_0 – номер по списку группы.

```
: image = cv2.imread('img/img.png')  
  
: result = cv2.morphologyEx(image, cv2.MORPH_TOPHAT, (52,52))  
  
: plt.figure(figsize=(10,10))  
  plt.subplot(121)  
  plt.imshow(image, cmap="gray")  
  plt.title("Оригинал")  
  plt.axis('off')  
  
  plt.subplot(122)  
  plt.imshow(result, cmap="gray")  
  plt.title("Цилиндр")  
  plt.axis('off')  
  
plt.show();
```

Оригинал



Цилиндр



Задание 8.6.

Применить операцию черная шляпа к изображению, размер ядра равен $40 + N_0$, N_0 – номер по списку группы.

```
: result = cv2.morphologyEx(image, cv2.MORPH_BLACKHAT, (80,80))  
  
: plt.figure(figsize=(10,10))  
  plt.subplot(121)  
  plt.imshow(image, 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();
```

Оригинал



Черная шляпа



Задание 8.7.

Изготовить ядро, его размер выбрать из ряда 3x3, 3x5, 5x3, 5x5, 5x7, 3x7, 7x3, 7x5, 5x7, 7x7, номер варианта должен быть равен номеру по списку группы (3x5). Обработать изображение с помощью выбранного ядра и ядра размером 9x9. Сравнить результаты обработки изображения этими ядрами.

```
image = cv2.imread('img/cat.jpg', 0)
img = cv2.resize(image, (900, 600))
```

```
kernel1 = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 5))
kernel2 = cv2.getStructuringElement(cv2.MORPH_RECT, (9, 9))
```

```
result1 = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel1)
result2 = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel2)
```

```
plt.figure(figsize=(10,10))
plt.subplot(121)
plt.imshow(result1, cmap="gray")
plt.title("Result 3x5")
plt.axis('off')
```

```
plt.subplot(122)
plt.imshow(result2, cmap="gray")
plt.title("Result 9x9")
plt.axis('off')
```

```
plt.show();
```

Result 3x5



Result 9x9



7. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.).

Условие. Обработать цветное изображение, применив к нему операции сглаживания, выделения границ, преобразования в полутоновое изображение, бинаризации и морфологического закрытия, а затем отобразить результаты.

Применение операции сглаживания

```
: image = cv2.imread('img/lis.jpg')

: # Сглаживание с использованием ядра размером 5x5
  blurred = cv2.blur(image, (5, 5))

: plt.figure(figsize=(10,10))
  plt.subplot(121), plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)), plt.title('Original Image')
  plt.axis('off')
  plt.subplot(122), plt.imshow(cv2.cvtColor(blurred, cv2.COLOR_BGR2RGB)), plt.title('Blurred')
  plt.axis('off')

  plt.tight_layout()
  plt.show();
```

Original Image



Blurred



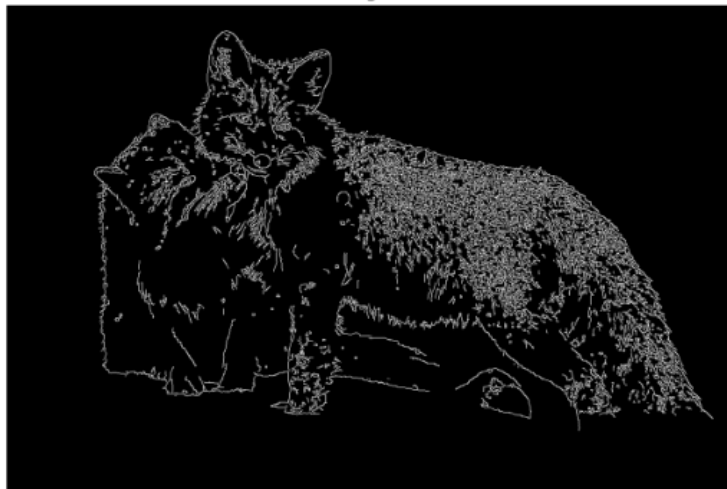
Применение операции выделения границ к исходному цветному изображению с помощью функции cv2.Canny().

```
: # Выделение границ
  edges = cv2.Canny(image, 100, 200)

: plt.imshow(edges, cmap='gray')
  plt.title('Edges')
  plt.axis('off')

  plt.tight_layout()
  plt.show();
```

Edges



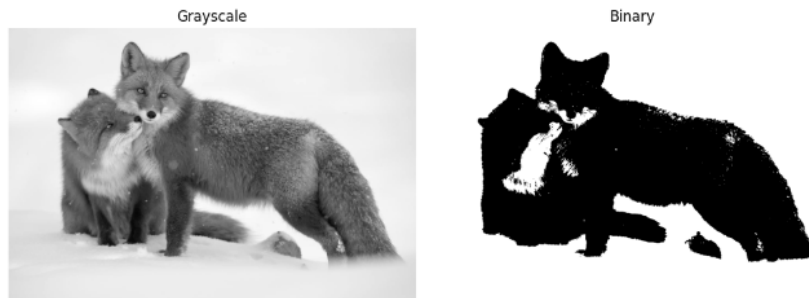
Применение операции бинаризации к полутоновому изображению с пороговым значением 170.

```
# Преобразование в полутоновое изображение
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Бинаризация
_, binary = cv2.threshold(gray_image, 170, 255, cv2.THRESH_BINARY)
```

```
plt.figure(figsize=(10,10))
plt.subplot(121), plt.imshow(gray_image, cmap='gray'), plt.title('Grayscale')
plt.axis('off')
plt.subplot(122), plt.imshow(binary, cmap='gray'), plt.title('Binary')
plt.axis('off')

plt.tight_layout()
plt.show();
```



Объединение результатов операций выделения границ edges и бинаризации binary с помощью операции побитового И с использованием функции cv2.bitwise_and(). Применение операции морфологического закрытия к объединенному изображению combined с использованием ядра размером 7x7.

```
# Объединение результатов выделения границ и бинаризации
combined = cv2.bitwise_and(edges, binary)

# Морфологическое закрытие с использованием ядра размером 7x7
kernel = np.ones((7, 7), np.uint8)
closed = cv2.morphologyEx(combined, cv2.MORPH_CLOSE, kernel)
```

```
plt.imshow(closed, cmap='gray'), plt.title('Closed')
plt.axis('off')
plt.tight_layout()
plt.show();
```



8. Зафиксируйте сделанные изменения в репозитории.

9. Выполните слияние ветки для разработки с веткой main(master).

10. Отправьте сделанные изменения на сервер GitHub.