

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

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе № 3.2**  
**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)

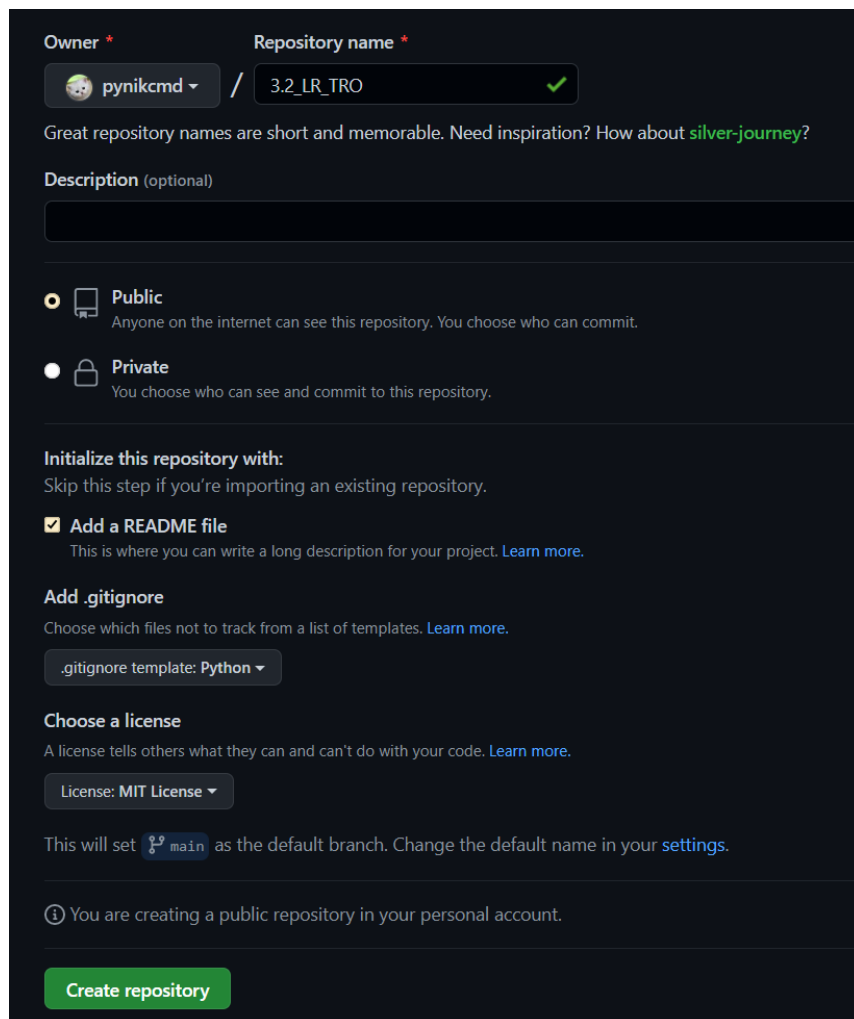
Ставрополь 2023

Цель работы: исследовать базовые возможности библиотеки NumPy языка программирования Python.

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 repository' form. At the top, the 'Owner' is set to 'pynikcmd' and the 'Repository name' is '3.2\_LR\_TRO', which is marked with a green checkmark. Below this, a message suggests that great repository names are short and memorable, with a link to 'silver-journey?'. The 'Description' field is optional and currently empty. Under the 'Visibility' section, the 'Public' option is selected, indicating that anyone on the internet can see the repository. The 'Private' option is also visible. The 'Initialize this repository with:' section includes a checkbox for 'Add a README file', which is checked. Below this, there is a section for 'Add .gitignore' with a dropdown menu set to 'Python'. The 'Choose a license' section has a dropdown menu set to 'MIT License'. At the bottom, a message states that the default branch will be set to 'main'. A green 'Create repository' button is at the bottom right.

Рисунок 1 – Создание репозитория

3. Выполните клонирование созданного репозитория на рабочий компьютер.

```
D:\fgit>git clone https://github.com/pynikcmd/3.2_LR_TR0.git
Cloning into '3.2_LR_TR0'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 13 (delta 3), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (13/13), 4.04 KiB | 826.00 KiB/s, done.
Resolving deltas: 100% (3/3), done.
```

Рисунок 2 – Клонирование репозитория

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

```
D:\fgit>git flow init
Initialized empty Git repository in D:/fgit/.git/
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
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/.git/hooks]
```

Рисунок 3 – Модель ветвления git-flow

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

6. Проработать примеры лабораторной работы.

7. Решить задания в ноутбуках, выданных преподавателем.

8. Создать ноутбук, в котором выполнить решение индивидуального задания. Ноутбук должен содержать условие индивидуального задания.

При решении индивидуального задания не должны быть использованы условный оператор if, а также операторы циклов while и for, а только средства библиотеки NumPy. Привести в ноутбуке обоснование принятых решений. Номер варианта индивидуального задания необходимо уточнить у преподавателя.

Условие. Вариант – 8.

. Соседями элемента  $A_j$  в матрице назовем элементы  $A_k$  с  $i - 1 < k < i + 1$ ,  $j - 1 < k < j + 1$ ,  $(k, 1)/(i, j)$ . Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы. Построить результат сглаживания заданной вещественной матрицы размером 7 на 7. В сглаженной матрице найти сумму модулей элементов, расположенных ниже главной диагонали.

```
In [74]: # Проверка на крайние нулевые столбцы и строки для корректного среза
for (i,j), element in np.ndenumerate(matrix):
    if i == 0 and j == 0:
        m = matrix[0:i+2, 0:j+2]
        rez = (np.sum(m) - element)/(m.size - 1)

    elif i == 0 and j != 0:
        m = matrix[0:i+2, j-1:j+2]
        rez = (np.sum(m) - element)/(m.size - 1)

    elif j == 0 and i != 0:
        m = matrix[i-1:i+2, 0:j+2]
        rez = (np.sum(m) - element)/(m.size - 1)

    else:
        m = matrix[i-1:i+2, j-1:j+2]
        rez = (np.sum(m) - element)/(m.size - 1)

    sglaj_matrix[i,j] = round(rez,2)

print("Сглаженная матрица:\n", sglaj_matrix)
```

```
Сглаженная матрица:
[[-13.   -4.8  -12.4  -6.2  -7.4  -9.6  -6.33]
 [-6.6  -3.12 -9.12  -4.5  -9.5  -6.88 -8.6 ]
 [-3.4  -4.   -5.5  -2.25 -10.  -8.38 -12.2 ]
 [-2.2  -1.75 -1.   -3.5  -5.5  -4.75 -9.8 ]
 [ 2.6  -1.25 1.88  -0.88 -6.25 -5.5  -12.4 ]
 [ 2.6  -0.62 -2.12 -2.62 -4.88 -4.   -6.   ]
 [ 5.    0.6   1.6  -3.8  -6.2  -4.8 -10.33]]
```

Рисунок 4 – Код программы

9. Зафиксируйте сделанные изменения в репозитории.

10. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.), условие которой предварительно необходимо согласовать с преподавателем.

Решение экспоненциальной функции вида:

$$F(x, y) = \exp(-x^2 - y^2)$$

```
In [35]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

# создание рисунка размером 15 на 15 см
inch = 2.54 # дюйм в см
fig1 = plt.figure(figsize=(15.0/inch, 15.0/inch))
ax1 = fig1.add_subplot(111)

# подписи осей на панели
ax1.set_xlabel(r'$x$')
ax1.set_ylabel(r'$y$')

# массив x-координат - 50 точек в диапазоне от -2 до 2
x = np.linspace(-2.0, 2.0, 50)

# массив y-координат - 50 точек в диапазоне от -2 до 2
y = np.linspace(-2.0, 2.0, 50)

# матрицы (сетка) координат
xx, yy = np.meshgrid(x, y)

# вычисление значений функции на сетке
F = np.exp(-xx**2 - yy**2)

# отображение 10 изолиний величины F
Gr = ax1.contour(xx, yy, F, 10)

# добавление подписей изолиний
# с помощью обращения к полю levels (списку изолиний
# на графике CS1) подписи выводятся только
# для каждой второй линии
ax1.clabel(Gr, Gr.levels[::2])
```

Рисунок 5 – Код программы

11. Зафиксируйте сделанные изменения в репозитории.

Контрольные вопросы

### 1. Каково назначение библиотеки NumPy?

NumPy — это библиотека для языка программирования Python, которая предоставляет в распоряжение разработчика инструменты для эффективной работы с многомерными массивами и высокопроизводительные вычислительные алгоритмы.

### 2. Что такое массивы ndarray?

Ndarray — это (обычно фиксированный размер) многомерный контейнер элементов одного типа и размера. Количество измерений и элементов в массиве определяется его формой, которая является кортежем из  $N$  натуральных чисел, которые определяют размеры каждого измерения.

### 3. Как осуществляется доступ к частям многомерного массива?

Элементы матрицы с заданными координатами: `m[1,0]`

Строка матрицы: `m[1, :]`

Столбец матрицы: `m[:, 1]`

Часть строки матрицы: `m[1, 2:]`

Часть столбца матрицы: `m[0:2, 1]`

Непрерывная часть матрицы: `m[0:2, 1:3]`

Произвольные столбцы / строки матрицы: `cols = [0, 1, 2]; m[:, cols]`

### 4. Как осуществляется расчет статистик по данным?

Размерность массива: `m.shape`

Вызов функции расчёта статистики: `m.max()`

Расчёт статистики по строкам или столбцам массива: `m.max(axis=1);`  
`m.max(axis=0)`

Индексы элементов с максимальным значением (по осям): `argmax`

Индексы элементов с минимальными значением (по осям): `argmin`

Максимальные значения элементов (по осям): `max`

Минимальные значения элементов (по осям): `min`

Средние значения элементов (по осям): `mean`

Произведение всех элементов (по осям): `prod`

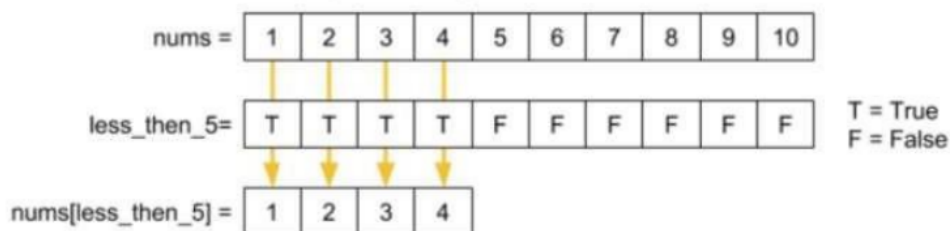
Стандартное отклонение (по осям): `std`

Сумма всех элементов (по осям): `sum` Дисперсия (по осям): `var`

## 5. Как выполняется выборка данных из массивов `ndarray`

```
>>> less_than_5 = nums < 5
>>> less_than_5
array([ True,  True,  True,  True, False, False, False, False, False,
        False])
```

Если мы переменную `less_than_5` передадим в качестве списка индексов для `nums`, то получим массив, в котором будут содержаться элементы из `nums` с индексами равными `True` позициям массива `less_than_5`, графически это будет выглядеть так.



```
>>> nums[less_than_5]
array([1, 2, 3, 4])
```