

## Лабораторная работа №2

### Создание сигнализационной системы на языке Python

#### Цель работы:

Изучить принципы анализа изображений и выявления различий с использованием библиотеки алгоритмов компьютерного зрения *OpenCV*.

#### Порядок работы:

1. Импорт необходимых библиотек.

```
from skimage.metrics import structural_similarity
import time
import cv2
import telebot
```

2. Подготовка токена и телеграм-id для работы с ботом.

```
token = '6504844786:AAH20gQCyOeocRuq676WuWniheTN7ruCVgY'
ID = '577314281'
bot = telebot.TeleBot(token)
```

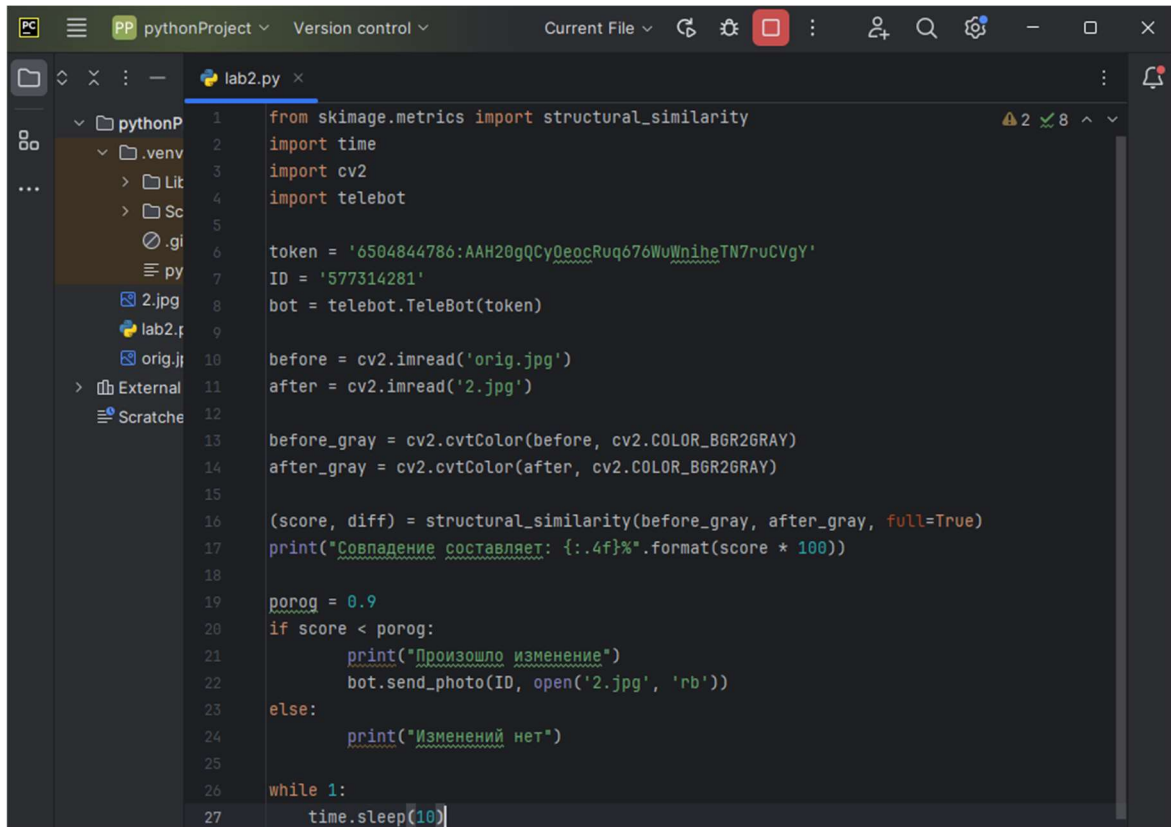
3. Считывание изображений и преобразование изображений в оттенки серого.

```
before = cv2.imread('orig.jpg')
after = cv2.imread('2.jpg')
before_gray = cv2.cvtColor(before, cv2.COLOR_BGR2GRAY)
after_gray = cv2.cvtColor(after, cv2.COLOR_BGR2GRAY)
```

4. Вычисление индекса структурного сходства между двумя изображениями и отправка изображения в телеграм-бот.

```
(score, diff) = structural_similarity(before_gray, after_gray, full=True)
print("Совпадение составляет: {:.4f}%".format(score * 100))
porog = 0.9
if score < porog:
    print("Произошло изменение")
    bot.send_photo(ID, open('2.jpg', 'rb'))
else:
    print("Изменений нет")
while 1:
    time.sleep(10)
```

5. Код программы представлен на рисунке 1.

The image shows a screenshot of a code editor window with a dark theme. The file explorer on the left shows a project structure with folders like 'pythonP', '.venv', 'Lit', 'Sc', and files like '2.jpg', 'lab2.py', 'orig.jpg'. The main editor area displays the code for 'lab2.py'. The code imports 'skimage.metrics' for 'structural\_similarity', 'time', 'cv2', and 'telebot'. It sets a 'token' and an 'ID', creates a 'bot' object, and reads two images ('orig.jpg' and '2.jpg'). These images are converted to grayscale. The 'structural\_similarity' function is used to compare the two grayscale images. A threshold 'porog' is set to 0.9. If the similarity score is below the threshold, a message is printed and a photo is sent via Telegram. Otherwise, a message is printed. A 'while' loop keeps the program running with a 10-second sleep interval.

```
1 from skimage.metrics import structural_similarity
2 import time
3 import cv2
4 import telebot
5
6 token = '6504844786:AAH20gQCy0eocRuq676WuWniheTN7ruCVgY'
7 ID = '577314281'
8 bot = telebot.TeleBot(token)
9
10 before = cv2.imread('orig.jpg')
11 after = cv2.imread('2.jpg')
12
13 before_gray = cv2.cvtColor(before, cv2.COLOR_BGR2GRAY)
14 after_gray = cv2.cvtColor(after, cv2.COLOR_BGR2GRAY)
15
16 (score, diff) = structural_similarity(before_gray, after_gray, full=True)
17 print("Совпадение составляет: {:.4f}%".format(score * 100))
18
19 porog = 0.9
20 if score < porog:
21     print("Произошло изменение")
22     bot.send_photo(ID, open('2.jpg', 'rb'))
23 else:
24     print("Изменений нет")
25
26 while 1:
27     time.sleep(10)
```

Рисунок 1 – Код программы

6. Проверка работы программы осуществляется при помощи изображений с камеры видеонаблюдения, представленных на рисунках 2-3.



Рисунок 2 – Исходное изображение

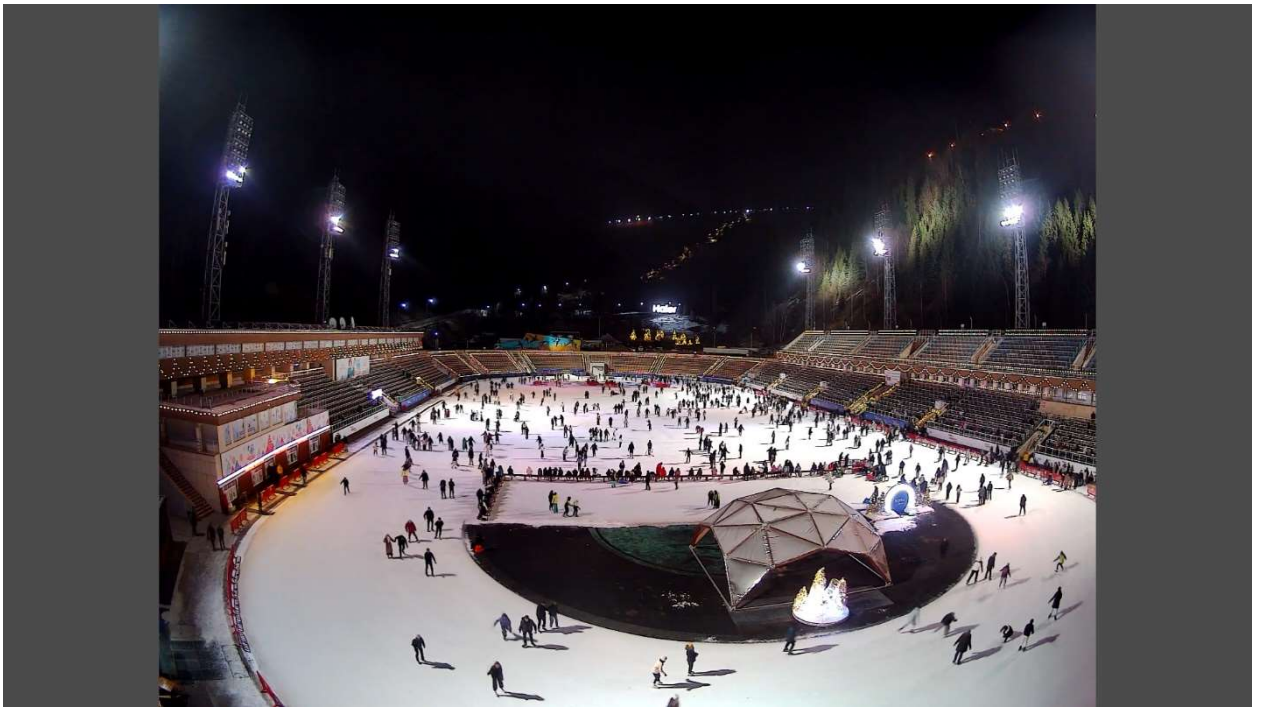


Рисунок 3 – Измененное изображение

7. Результат работы программы представлен на рисунках 4-5.



Рисунок 4 – Результат работы программы

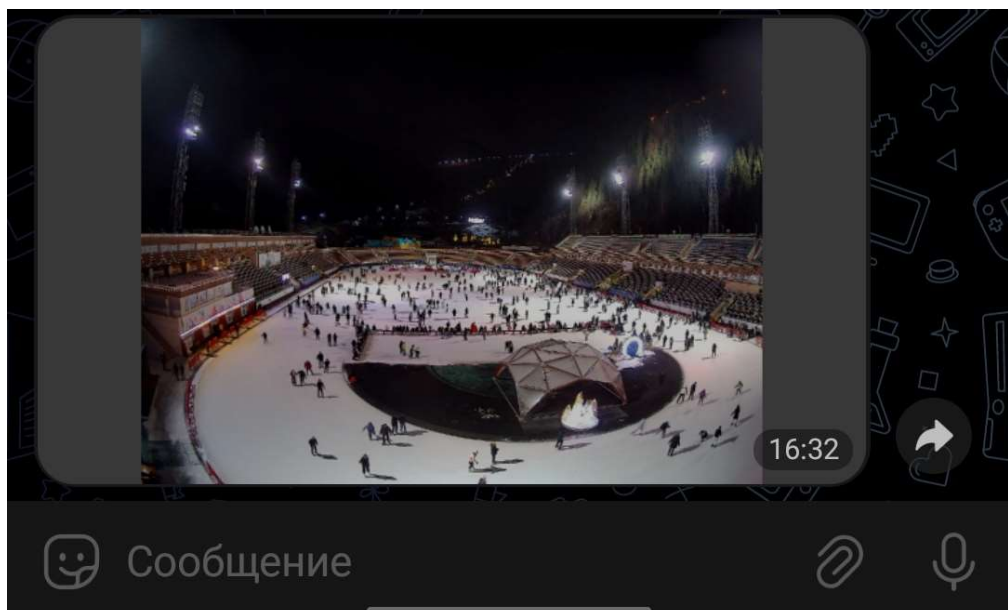


Рисунок 5 – Отправленное в телеграм-бот изображение

8. Модифицировал программу для считывания изображения с внешней веб-камеры.

```
from skimage.metrics import structural_similarity
import time
import cv2
import telebot

token = '6504844786:AAH20gQCyOeocRuq676WuWniheTN7ruCVgY'
ID = '577314281'
bot = telebot.TeleBot(token)

cam = cv2.VideoCapture(0)
ret, image = cam.read()
cam.release()
cv2.imwrite('filename.jpg', image)
img1 = cv2.imread('filename.jpg')
before_gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
while 1:
    print("I am listening...")
    time.sleep(1)
    cam = cv2.VideoCapture(0)
    ret, image = cam.read()
    cv2.imwrite('newpic.jpg', image)
    img2 = cv2.imread('newpic.jpg')
    after_gray = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
    (score, diff) = structural_similarity(before_gray, after_gray, full=True)
    print("Совпадение составляет: {:.4f}%".format(score * 100))
    if score < 0.9:
        print("Произошло изменение")
        bot.send_photo(ID, open('newpic.jpg', 'rb'))
    else:
        print("Изменений нет")
    time.sleep(1)
```

Результаты представлены на рисунках 7-8.

```
Run lab2 x lab_2_cam x
C:\labs\iis\pythonProject\.venv\Scripts\python.exe C:\labs\iis\pythonProject\lab_2_cam.py
I am listening...
Совпадение составляет: 98.4768%
Изменений нет
I am listening...
Совпадение составляет: 98.2176%
Изменений нет
I am listening...
Совпадение составляет: 85.2504%
Произошло изменение
I am listening...
Совпадение составляет: 98.2746%
Изменений нет
I am listening...
Совпадение составляет: 94.2696%
Изменений нет
I am listening...
Traceback (most recent call last):
  File "C:\labs\iis\pythonProject\lab_2_cam.py", line 18, in <module>
    cam = cv2.VideoCapture(0)
          ^^^^^^^^^^^^^^^^^^^^^
KeyboardInterrupt

Process finished with exit code -1073741510 (0xC000013A: interrupted by Ctrl+C)
```

Рисунок 7 – Выполнение программы

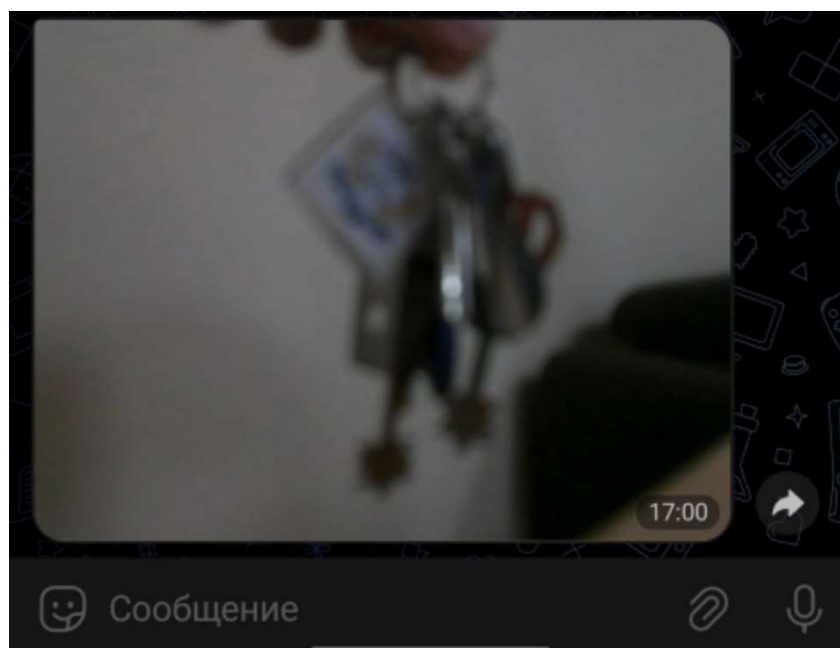


Рисунок 8 – Полученное изображение

**Вывод:** получили опыт работы с библиотекой алгоритмов компьютерного зрения *OpenCV*, с библиотекой *Telepot* для работы с *Telegram Bot API*, создали программу для определения различий между похожими изображениями с функцией оповещения посредством отправки снимка с камеры в *Telegram*-бот.