**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.

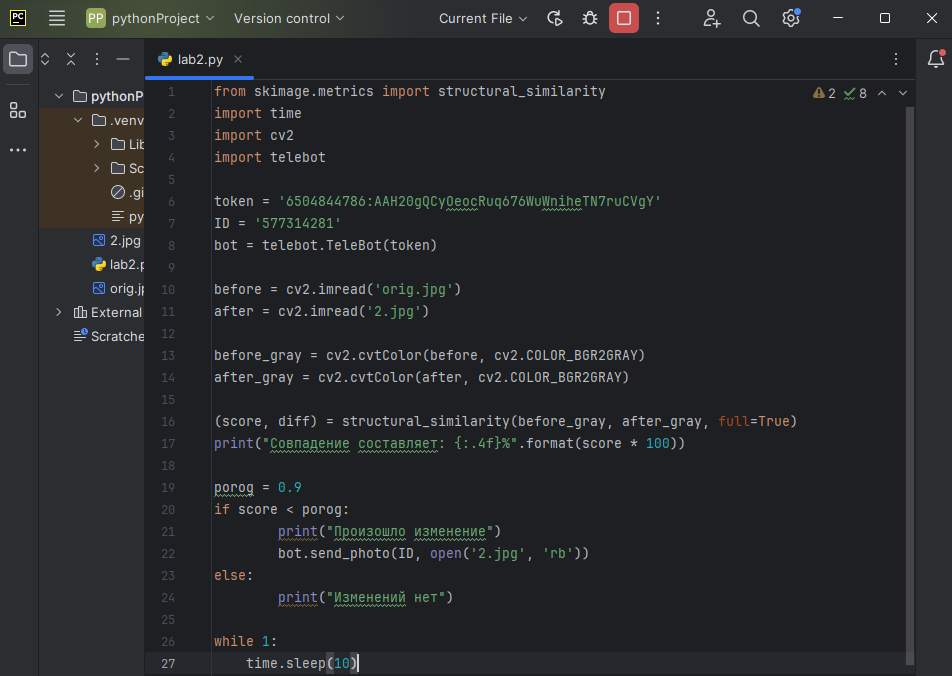


Рисунок 1 – Код программы

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.

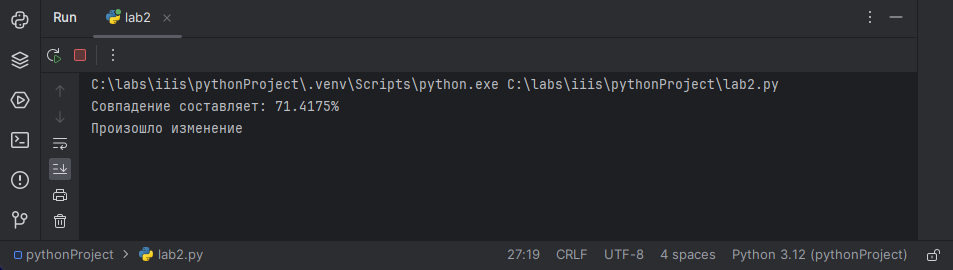


Рисунок 4 – Результат работы программы

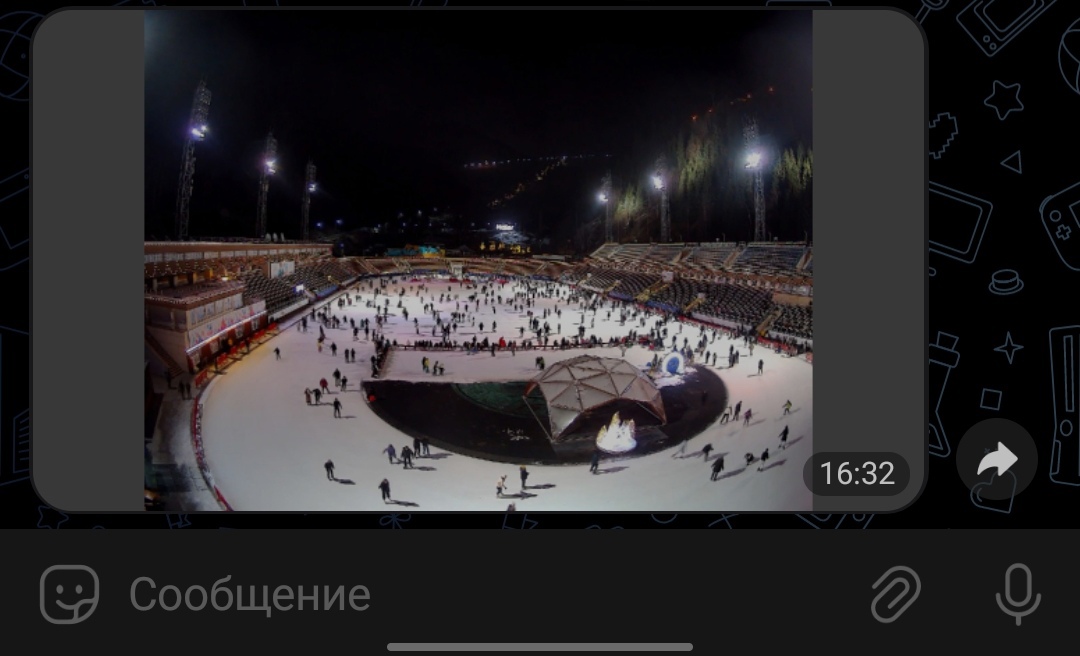


Рисунок 5 – Отправленное в телеграм-бот изображение

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.

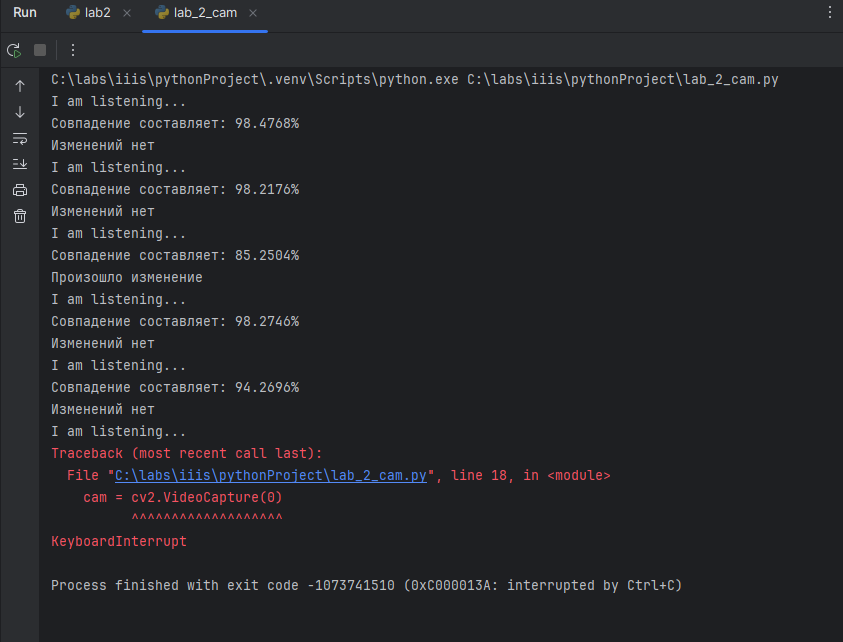


Рисунок 7 – Выполнение программы

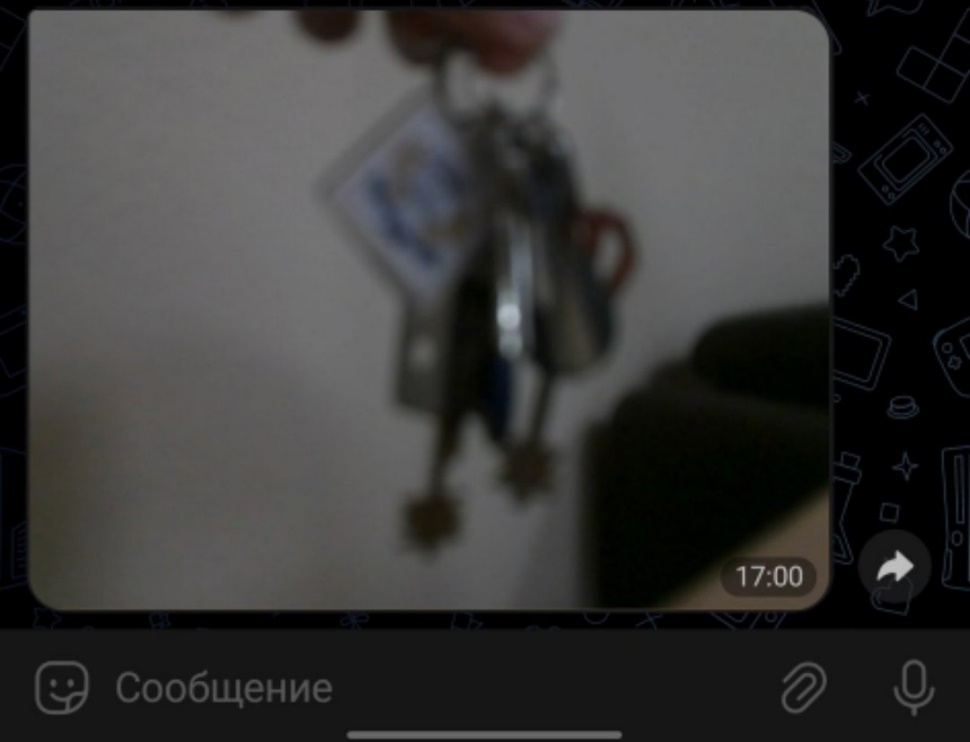


Рисунок 8 – Полученное изображение

**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.