

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 = '6653814498:AAG39oz3LnLMh0OEzbJ3fefKVAKHc8k4mDs'
```

3. ID = '407359081'

```
bot = telebot.TeleBot(token)
```

4. Считывание изображений и преобразование изображений в оттенки серого.

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

5. 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 = 0.95
if score < porog:
    print("Произошло изменение")
    bot.send_photo(ID, open('2.jpg', 'rb'))
else:
    print("Изменений нет")
while 1:
    time.sleep(10)
```

6. Код программы представлен на рисунке 1.

```
from skimage.metrics import structural_similarity
import time
import cv2
import telebot
token = '6653814498:AAG39oz3LnLMh00EzbJ3fefKVAKHc8k4mDs'
ID = '407359081'
bot = telebot.TeleBot(token)

before = cv2.imread('origin.jpg')
after = cv2.imread('mod.jpg')
before_gray = cv2.cvtColor(before, cv2.COLOR_BGR2GRAY)
after_gray = cv2.cvtColor(after, cv2.COLOR_BGR2GRAY)
(score, diff) = structural_similarity(before_gray, after_gray, full=True)
print("Совпадение составляет: {:.4f}%".format(score * 100))
porog = 0.95
if score < porog:
    print("Произошло изменение")
    bot.send_photo(ID, open('mod.jpg', 'rb'))
else:
    print("Изменений нет")
```

Рисунок 1 – Код программы

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



Рисунок 2 – Исходное изображение



Рисунок 3 – Измененное изображение

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

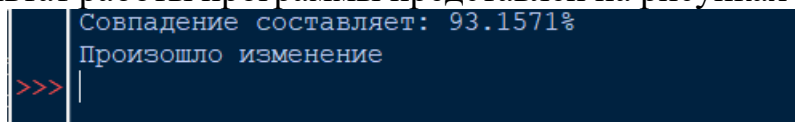


Рисунок 4 – Результат работы программы



Рисунок 5 – Отправленное в телеграм-бот 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 в *Telegram*-бот.