# 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:

## Импорт необходимых библиотек.

import cv2 import telepot

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

## **token = '6883457272:AAGw7YuYDQ3yDEkQcsxW6Xhq3NgVYEdfU58'**

## **ID = '5330919200'**

## **bot = telepot.Bot(token)**

## Считывание изображений и анализ.

**#Функция вычисления хэша**

**def CalcImageHash(FileName):**

**image = cv2.imread(FileName) #Прочитаем картинку**

**resized = cv2.resize(image, (8,8), interpolation = cv2.INTER\_AREA) #Уменьшим картинку**

**gray\_image = cv2.cvtColor(resized, cv2.COLOR\_BGR2GRAY) #Переведем в черно-белый формат**

**avg=gray\_image.mean() #Среднее значение пикселя**

**ret, threshold\_image = cv2.threshold(gray\_image, avg, 255, 0) #Бинаризация по порогу**

**#Рассчитаем хэш**

**\_hash=""**

**for x in range(8):**

**for y in range(8):**

**val=threshold\_image[x,y]**

**if val==255:**

**\_hash=\_hash+"1"**

**else:**

**\_hash=\_hash+"0"**

**return \_hash**

**def CompareHash(hash1,hash2):**

**l=len(hash1)**

**i=0**

**count=0**

**while i<l:**

**if hash1[i]!=hash2[i]:**

**count=count+1**

**i=i+1**

**return count**

**hash1=CalcImageHash("Game")**

**hash2=CalcImageHash("Game1")**

1. Вычисление разницы между изображениями и отправка изображения в телеграм-бот.

## **if CompareHash(hash1, hash2)>6:**

## **print("На захваченном камерой и сохранённом внешним ПО фото ЧТО-ТО ПОМЕНЯЛОСЬ !")**

## **bot.sendPhoto(ID, open('Game1.jpg', 'rb'))**

## **else:**

## **print("всё по-старому. нет поводов для паники")**Код программы представлен на рисунке 1.

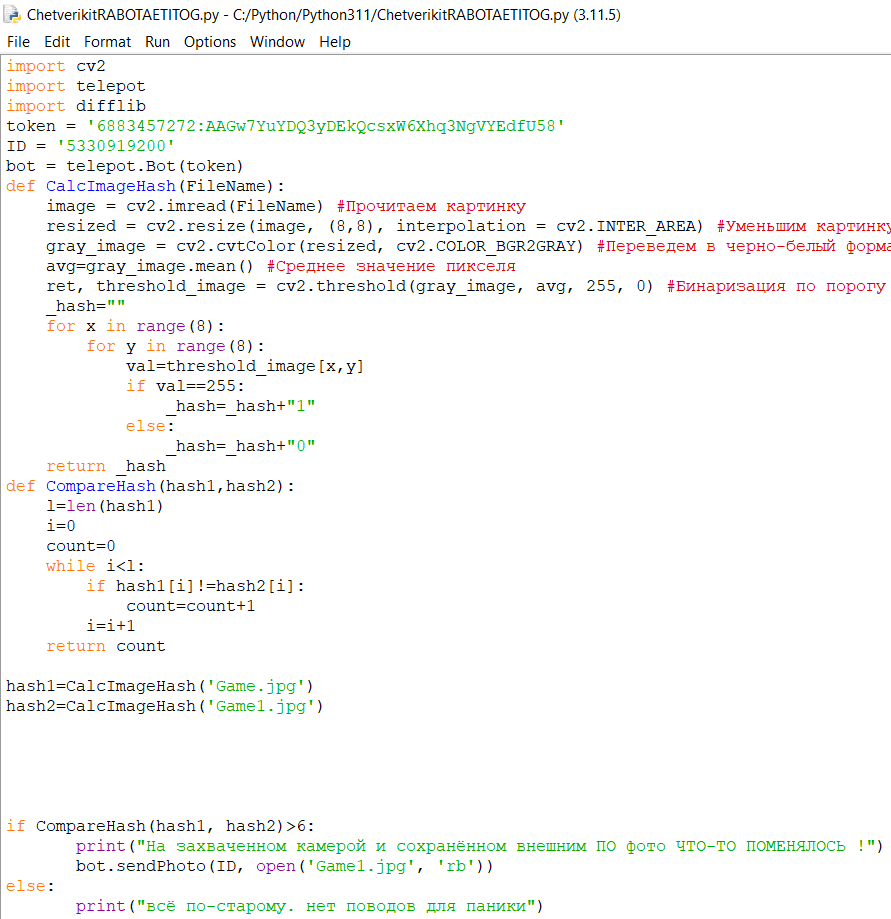


Рисунок 1 - Код программы

## Проверка работы программы осуществляется при помощи изображений с камеры видеонаблюдения, представленных на рисунках 2-4.



Рисунок 2 - Исходное изображение



Рисунок 3 – На кадре ещё ничего не поменялось



Рисунок 4 – Кадр изменился

## Результат работы программы представлен на рисунках 5-6.

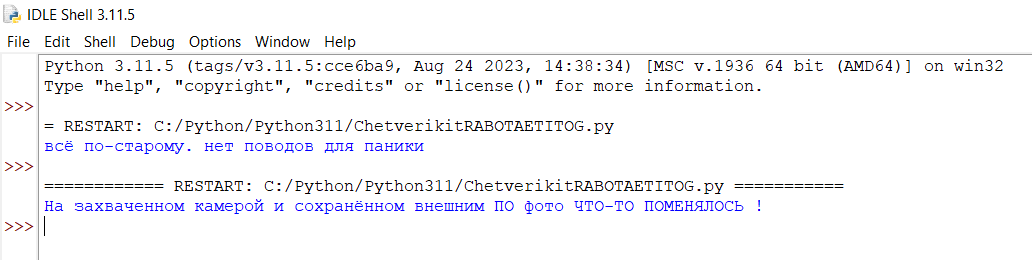


Рисунок 5 - Результат работы

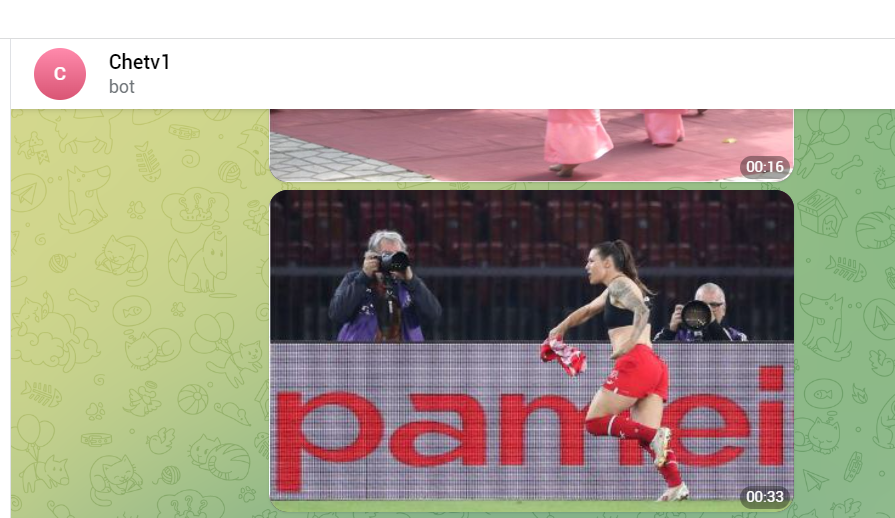


Рисунок 6 - Отправка фото в бот

Периодическое обновление снимка Game1 в папке можно организовать с помощью внешней программы, написанной в среде LabVIEW с помощью библиотеки Vision and Motion, используя незавершающийся цикл, в каждой итерации которого будет происходить захват изображения с камеры и сохранение его по адресу , в котором находится фото Game1. Созданный

В LabVIEW виртуальный прибор можно в этом же приложении преобразовать в exe формат. Тогда для его запуска понадобится установить только LabVIEW Run Time Engine, LabVIEW Run-Time Vision , которые весят значительно меньше среды LabVIEW.

**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.

Работу выполнил: Четверик И.С