|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИlogo | | | | | | | | | | | | | | | | | | | | |
| Федеральное государственное автономное образовательное  учреждение высшего образования | | | | | | | | | | | | | | | | | | | | |
| **Дальневосточный федеральный университет** | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК** | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| **Кафедра информационной безопасности** | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| **О Т Ч Е Т** | | | | | | | | | | | | | | | | | | | | |
| о прохождении учебной практики (учебно-лабораторного практикума) | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | |  |  | Выполнил студент  гр. С8117-10.05.01ммзи | | | | | | | | |
|  |  |  | | | | | | Шадрин С.А. | | |
|  | | | | | | | | | |  |  | (подпись) | | | | | |  | | |
|  | | | | | | | | | | | | | | | | | | | | |
| Отчет защищен с оценкой | | | | | | | | | |  |  | Руководитель практики | | | | | | |  | |
|  | | | | | | | | | |  |  | Старший преподаватель кафедры информационной безопасности ШЕН | | | | | | | | |
|  | | | |  | С.С. Зотов | | | | |  |  |  | | | | |  | С.С. Зотов | | |
| (подпись) | | | |  | (И.О. Фамилия) | | | | |  |  | (подпись) | | | | |  | (И.О. Фамилия) | | |
| « | 26 | » | июня | | | | | | 2021 г. |  |  |  | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| Регистрационный № | | | | | | | |  | |  |  | Практика пройдена в срок | | | | | | | | |
| « | 26 | » | июня | | | | | | 2021 г. |  |  | с | « | 22 | » | февраля | | | | 2021 г. |
|  | | | | | | | | | |  |  | по | « | 26 | » | июня | | | | 2021 г. |
|  | | | | | |  | Е.В. Третьяк | | |  |  | на предприятии | | | | | | | | |
| (подпись) | | | | | |  | (И.О. Фамилия) | | |  |  |
|  | | | | | | | | | |  |  | Кафедра информационной | | | | | | | | |
|  | | | | | | | | | |  |  | безопасности ШЕН ДВФ | | | | | | | | |
|  | | | | | | | | | |  |  |  | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| г. Владивосток | | | | | | | | | | | | | | | | | | | | |
| 2021 | | | | | | | | | | | | | | | | | | | | |

Оглавление

[**Задание на практику** 3](#_Toc75305085)

[**Введение** 4](#_Toc75305086)

[**Применения языка Python в реализации поставленной задачи** 5](#_Toc75305087)

[**Библиотеки face\_recognation и opencv2** 8](#_Toc75305088)

[**Разработка прототипа программы для распознавания лиц** 10](#_Toc75305089)

[**Заключение** 13](#_Toc75305090)

[**Список использованной литературы** 14](#_Toc75305091)

# **Задание на практику**

- Написание программы по анализу видеофайлов, которая подсчитывает количество людей на видео и сохраняет результат работы в виде изображений, содержащих лица участников видео без повторений.

- Написание отчета по практике о проделанной работе.

# **Введение**

Учебная практика (учебно-лабораторный практикум) проходила на кафедре информационной безопасности ШЕН ДВФУ в период с 22 февраля 2021 года по 26 июня 2021 года.

Целью прохождения практики является приобретение практических и теоретических навыков по специальности, а также навыков оформления проведенного исследования в отчетной форме.

Задачи практики:

1. Изучить основы языка Python, библиотеки необходимые для реализации программы.

2. Разработать программу для распознавания лиц с видеофайла и сохранения результата работы.

3. На основе полученных знаний написать отчет по практике о проделанной работе.

# **Применения языка Python в реализации поставленной задачи**

Для выполнения поставленной задачи, распознавания лиц в видеофайле и сохранения результатов работы программы был выбран язык программирования Python.

Python - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным — всё является объектами. Основными особенностями данного языка программирования являются: разделение блоков кода пробельными отступами; минималистичность синтаксиса языка; поддержка императивного, процедурного, структурного, объектно-ориентированного программирования; простота в изучении языка. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как Си или C++.

Причиной выбора данного языка программирования для выполнения поставленной задачи послужили простота в его изучении, множество библиотек, применяемых в различных направлениях программирования.

Для реализации программы были использованы следующие команды и примитивы языка:

1.Переменные

Необходимы для хранения данных, при объявлении тип данных не указывается, Python определяет его автоматически.

Пример :



2.Списки

Список представляет тип данных, который хранит набор или последовательность элементов. Для объявления нет необходимости указывать размерность или тип хранимых данных.

Пример:



3. Инструкция import

Данная инструкция используется для подключения пакетов, модулей (библиотек).

Пример:



4.Функция int()

Принимает в качестве аргумента строку и возвращает целочисленное значение, то есть если указать в качестве аргумента что-либо кроме целого числа произойдет ошибка во время исполнения кода.

Пример:



5.Метод get()

Возвращает значение для данного ключа. Если ключ недоступен, возвращается значение по умолчанию None.

Пример:



6.Метод read()

Считывает файл по указанному пути в переменную.

Пример:



7.Функция print()

Выводит в консоль все переданные ей аргументы в формате строки.

Пример:



8.Метод format()

Принимает произвольное количество аргументов и выполняет их подстановку в указанных местах строки, относительно которой он вызван

Пример:



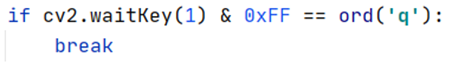
9.Циклы for и while

Цикл for применяется для выполнения части кода определенное количество итераций. Цикл while выполняется пока будет верным некоторое условие.

10.Инструкция if <>:….

Проверяет выполняется ли условие <>, если выполняется то исполняется вложенная в структуру часть кода.

Пример:



# **Библиотеки face\_recognation и opencv2**

OpenCV — это общедоступная библиотека «компьютерного зрения», которая предназначена для анализа, классификации и обработки изображений и видеофайлов.

Из ее основных возможностей можно отметить:

1. Кадрирование изображений

2. Изменение размера изображений

3. Поворот изображений

4. Перевод цветного изображения в черно-белое/ градации серого.

5. Размытие/ сглаживание изображений

6. Рисование прямоугольников и линий на изображениях

7. Надписи (текст на изображении).

В работе использовались следующие методы и функции из этой библиотеки:

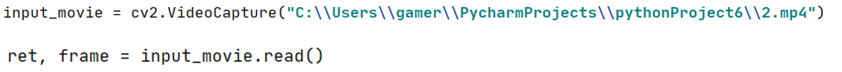
1. Функция VideoCapture()

Функция выполняет покадровое считывание из видеофайла, в качестве аргумента принимает путь к видеофайлу. При использовании с методом read() на выходе получаем два объекта :

1. Булевое значение (True или False), в случае отсутствия ошибок при загрузке текущего кадра - True.

2. Текущий прочитанный кадр из видео.

Пример:



2. Функция VideoWriter()

Функция служит для сохранения последовательности изображений в видеофайл, в качестве аргументов функция принимает название файла который будет создан, формат, частоту кадров, разрешение.

Пример:



face\_recognation - это общедоступная библиотека распознавания лиц на изображении или в видеоряде.

В ее основные функции входит:

1. Поиск лиц на фото.

2. Поиск черт лица на фотографиях.

Распознавание фото с получением контуров и важных для идентификации черт лица (глаза, нос, рот, подбородок).

3. Составление карты координат специфических лицевых точек.

Составление карты лица по 68 точкам, которые позволяют с большой точностью идентифицировать человека.

4. Идентификация лиц на фотографиях.

5. Распознавание лиц в потоковом видео (к примеру с веб-камеры)

В ходе работы была использована функция face\_locations(), которая распознает лица на изображении. В качестве аргумента принимает изображение (кадр из видео) а на выход дает массив координат, образующих рамки вокруг лиц, найденных на изображении.

Пример:



# **Разработка прототипа программы для распознавания лиц**

На основе прототипа программы, которая уже могла распознавать лица с веб камеры и видеофайлов, была получена финальная версия программы, которая научилась запоминать лица людей и различать их. Далее подробно рассмотрим код программы и результат ее работы.

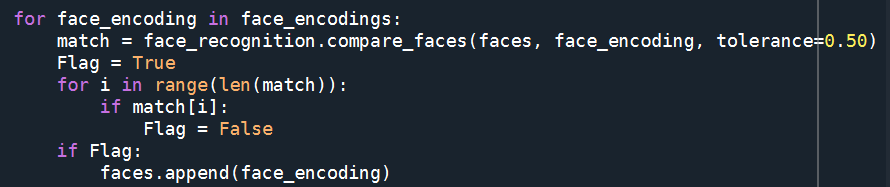
1. Анализируя видео покадрово, программа будет находить лица. Для того, чтобы сравнивать лица на каждом кадре с теми, что уже встречались, были созданы следующие массивы:

faces = []

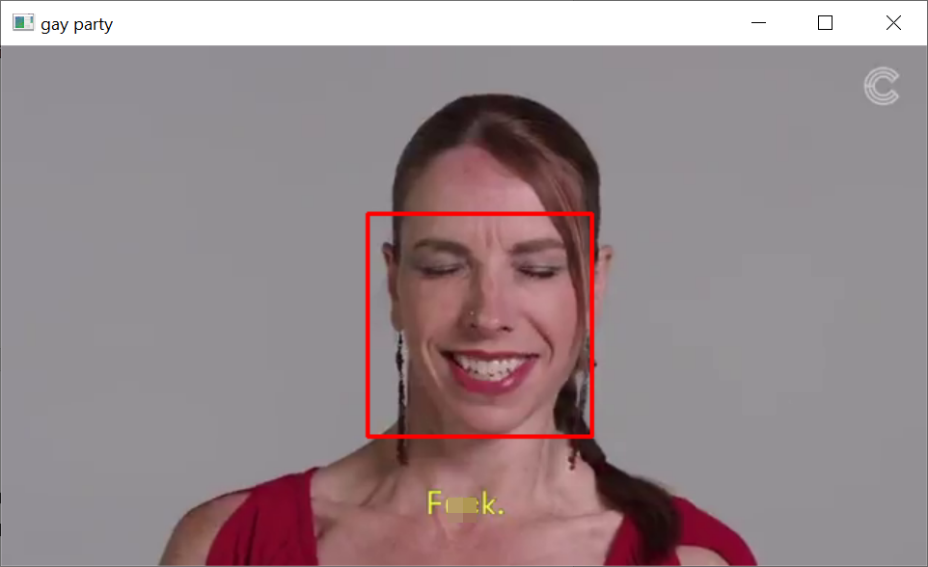
face\_locations = []

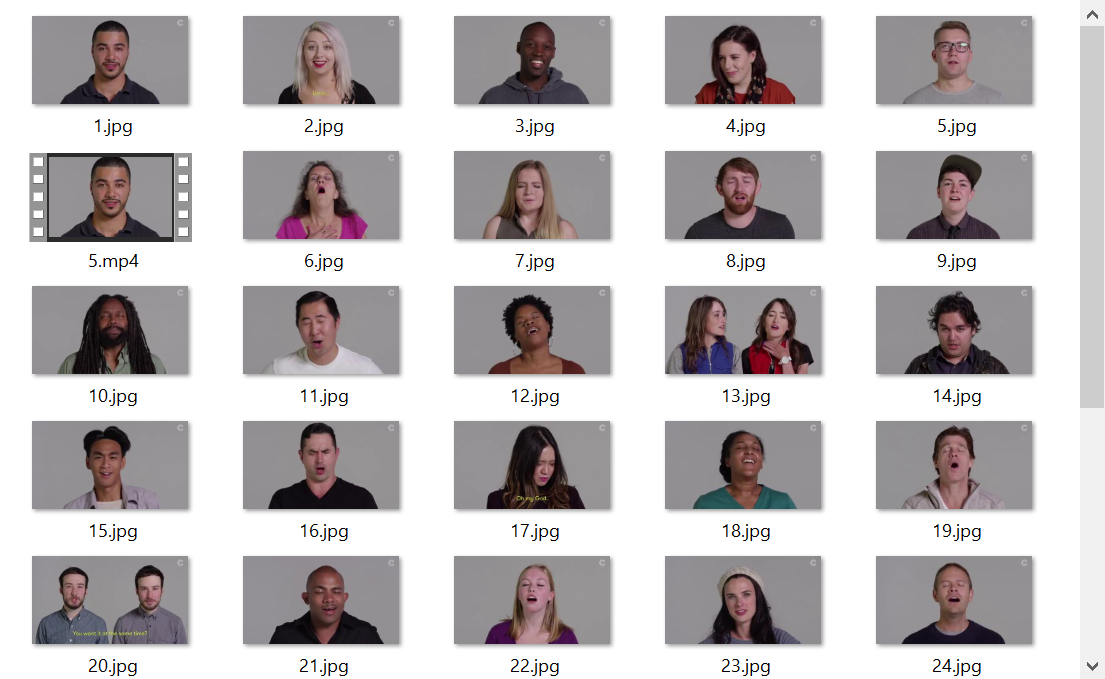
face\_encodings = []

1. Функция face\_recognition.face\_locations(rgb\_frame) находит на текущем кадре лицо. Эти данные запоминаются в массиве face\_locations. Функция face\_recognition.face\_encodings(rgb\_frame, face\_locations) получает на вход текущий кадр с координатами лица и получает вектор признаков для него, который заносится в массив face\_encodings. Массив необходим для того, чтобы запомнить все лица в кадре, если их больше, чем одно.
2. В следующем цикле прогоняются все лица, найденные в кадре. Функция face\_recognition.compare\_faces(faces, face\_encoding, tolerance=0.50) каждое найденное лицо с теми, что уже были встречены и занесены в массив faces с точностью 0.50. Если обнаружено совпадение, то текущее лицо не запоминается и пропускается, иначе заносится в массив faces.



1. Следующим шагом сохраняется текущий кадр в отдельном .jpg файле, если было найдено новое лицо. Название соответствует номеру лица в массиве faces, что также и является счётчиком уникальных лиц.





Финальный код:

import cv2

import face\_recognition

input\_movie = cv2.VideoCapture("C:\\temp\\5.mp4")

length = int(input\_movie.get(cv2.CAP\_PROP\_FRAME\_COUNT))

fourcc = cv2.VideoWriter\_fourcc(\*'XVID')

output\_movie = cv2.VideoWriter("C:\\temp\\output2.avi", fourcc, 23.98, (360, 640))

faces = []

face\_locations = []

face\_encodings = []

frame\_number = 0

while True:

ret, frame = input\_movie.read()

frame\_number += 1

if not ret:

break

rgb\_frame = frame[:, :, ::-1]

face\_locations = face\_recognition.face\_locations(rgb\_frame)

face\_encodings = face\_recognition.face\_encodings(rgb\_frame, face\_locations)

for face\_encoding in face\_encodings:

match = face\_recognition.compare\_faces(faces, face\_encoding, tolerance=0.60)

Flag = True

for i in range(len(match)):

if match[i]:

Flag = False

if Flag:

faces.append(face\_encoding)

cv2.imwrite("C:/temp/"+str(len(faces))+".jpg", frame)

for top, right, bottom, left in face\_locations:

cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

print("Writing frame {} / {}".format(frame\_number, length))

output\_movie.write(frame)

cv2.imshow('gay party', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

input\_movie.release()

output\_movie.release()

cv2.destroyAllWindows()

# **Заключение**

Для реализации поставленной задачи, в процессе прохождения учебной практики (учебно-лабораторного практикума) были изучены основы языка программирования Python, а также были изучены материалы по библиотекам OpenCV и Face Recognition.

Был реализован прототип программы для распознавания лиц в потоковом видео с веб-камеры, в последующем доработан для распознавания лиц с видеофайлов, а также добавлена функция сохранения результатов помимо видеофайла в отдельные скриншоты различных людей, встречающихся в видеофайле, без повторов.

В ходе прохождения практики все задачи были выполнены, а цель достигнута.

# **Список использованной литературы**

1. Face Recognition with Python, in Under 25 Lines of Code [Электронный ресурс]. – Электрон. дан. – Режим доступа: https://realpython.com/face-recognition-with-python/ (дата обращения: 27.04.2021)
2. face\_recognition package [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://face-recognition.readthedocs.io/en/latest/face_recognition.html> (дата обращения: (27.04.2021)
3. Face Recognition Python [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://github.com/ageitgey/face_recognition> (29.05.2021)
4. Самоучитель python [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://pythonworld.ru/samouchitel-python> (19.05.2021)
5. Open CV. Основы работы с видео [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://systop.ru/linux/42-open-cv-osnovy-raboty-s-video.html> (17.05.2021)
6. Шпаргалка по OpenCV — Python [Электронный ресурс]. – Электрон. дан. – Режим доступа: <https://tproger.ru/translations/opencv-python-guide/#faces> (21.05.2021)
7. OpenCV в Python. Часть 1 [Электронный ресурс]. – Электрон. дан. – Режим доступа: https://habr.com/ru/post/519454/ (01.06.2021)