МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ

ФЕДЕРАЦИИ

**Государственное образовательное бюджетное учреждение высшего**

**образования**

**«Кубанский Государственный Университет»**

**(ФГБОУ ВО «КубГУ»)**

Факультет компьютерных технологий и прикладной математики

**ОТЧЁТ**

**Обнаружение и распознавание лица на Python с использованием OpenCV**

Дисциплина: Нечёткие и нейросетевые технологии в экономике

Выполнил:

Студент 47 группы

Хачатрян Арутюн Арамович

Преподаватель:

Прутский Алексей Сергеевич

Краснодар

2019

# Введение

В данной работе мы будем использовать камеру компьютера для распознавания лица в режиме реального времени с помощью Python и OpenCV.

OpenCV – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Java, Ruby, Matlab, Lua и других языков. Может свободно использоваться в академических и коммерческих целях - распространяется в условиях лицензии BSD.

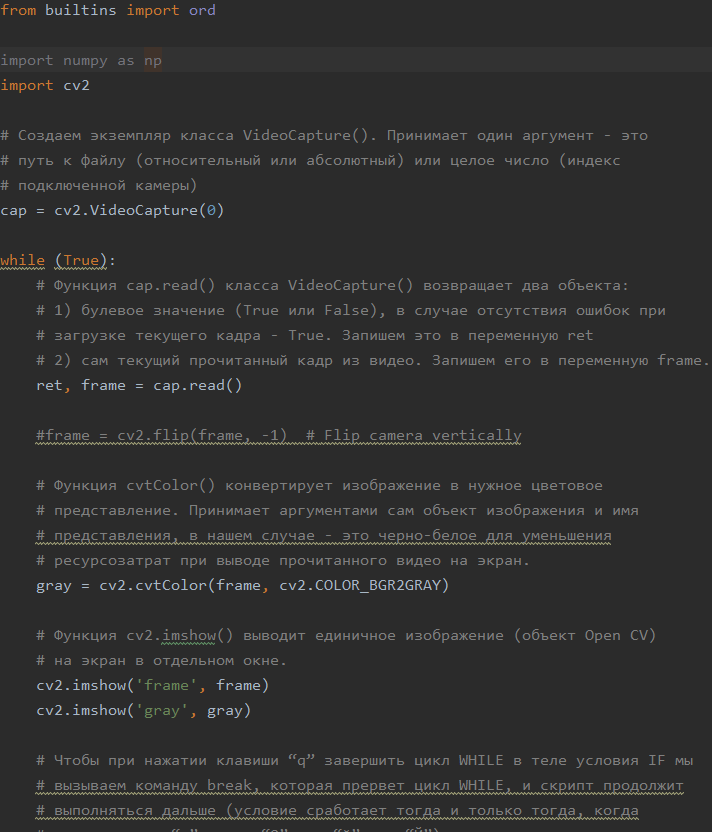
OpenCV был разработан для эффектного вычисления и работы в реальном времени. Таким образом, он идеально подходит для распознавания лиц в реальном времени с помощью камеры.

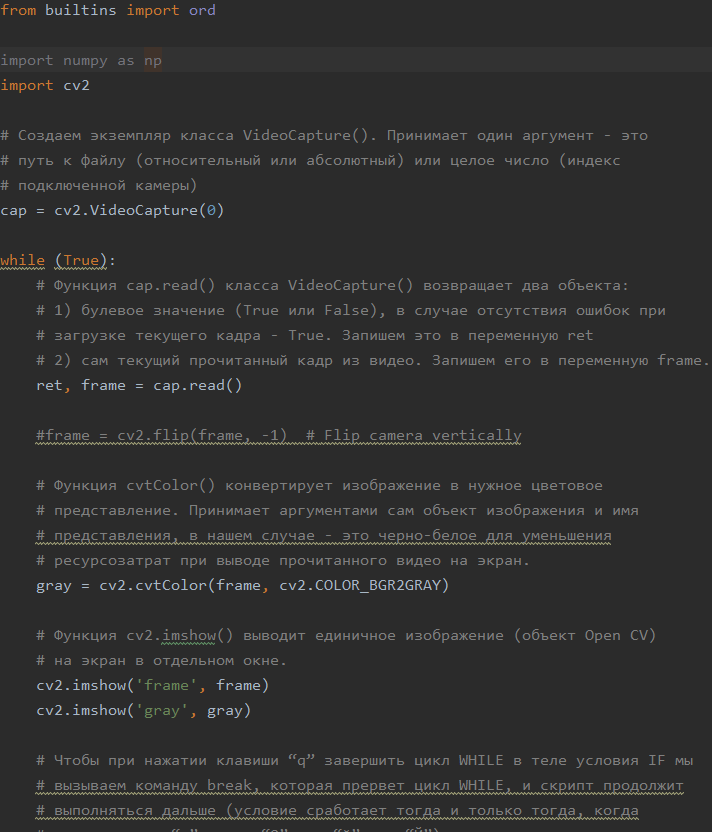
Чтобы создать полный проект по распознаванию лиц, мы должны работать на 3 очень разных этапах:

* Обнаружение лиц и сбор данных
* Обучение распознавателя
* Распознавание лица

## Тестирование камеры

Нижеприведенный код будет захватывать видеопоток, который будет создан вашей камерой, и отображать его как в цвете BGR, так и в режиме Gray. Чтобы закончить программу, вы должны нажать клавишу [q] на клавиатуре. На рисунке под приведённым кодом программы показан результат.





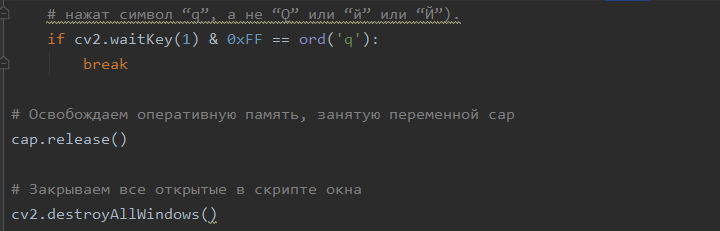


Рисунок 1 – Тестирование камеры



Рисунок 2 – Результат тестирования камеры

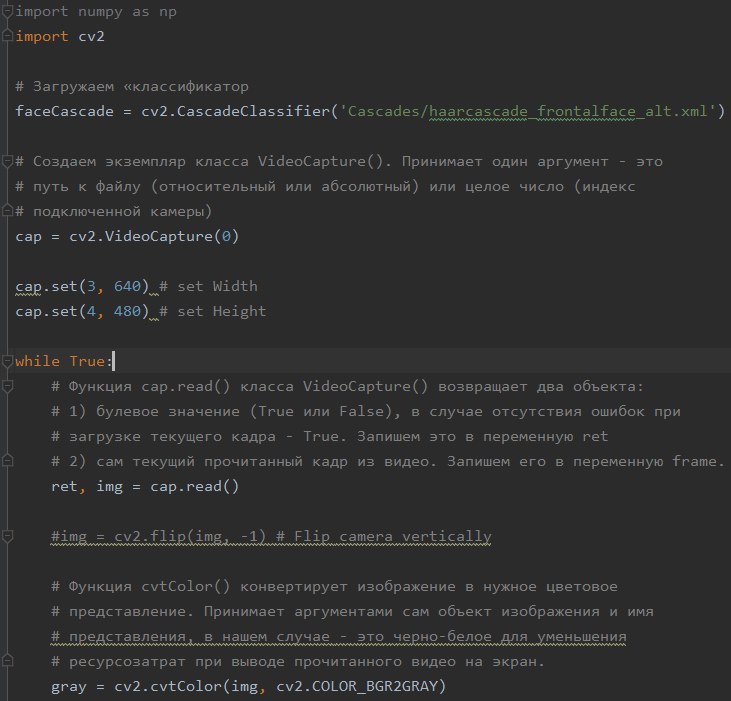
## Определение лица

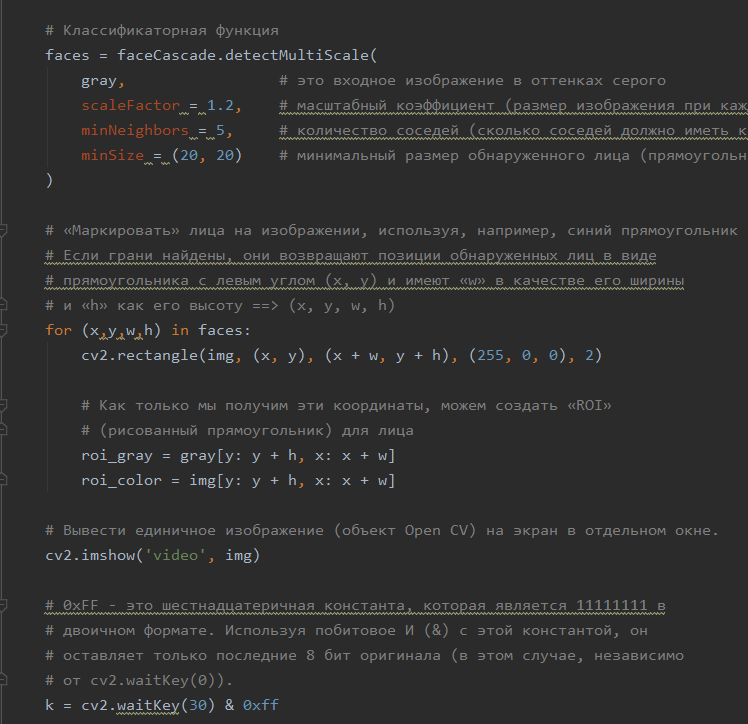
Самой основной задачей Распознавания Лица является, конечно же, «Определения Лица». Прежде всего, вы должны «захватить» лицо, чтобы распознать его, по сравнению с новым лицом, захваченным в будущем.

Наиболее распространенный способ обнаружения лица (или любых объектов), использует «классификатор каскадов Хаар».

Обнаружение объектов с использованием «классификатора каскадов Хаар» - эффективный метод обнаружения объектов, предложенный Полом Виолой и Майклом Джонсом в своей статье «Быстрое обнаружение объектов с использованием расширенного каскада простых функций» в 2001 году. Это подход, основанный на механизме обучения, каскадная функция обучается из множества положительных и отрицательных изображений. Затем он используется для обнаружения объектов на других изображениях.

OpenCV уже содержит много предварительно подготовленных классификаторов для лица, глаз, улыбки и т. д..





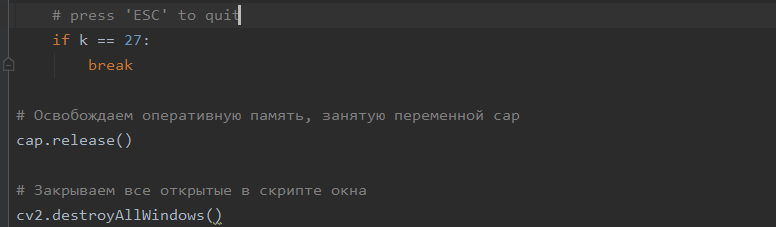
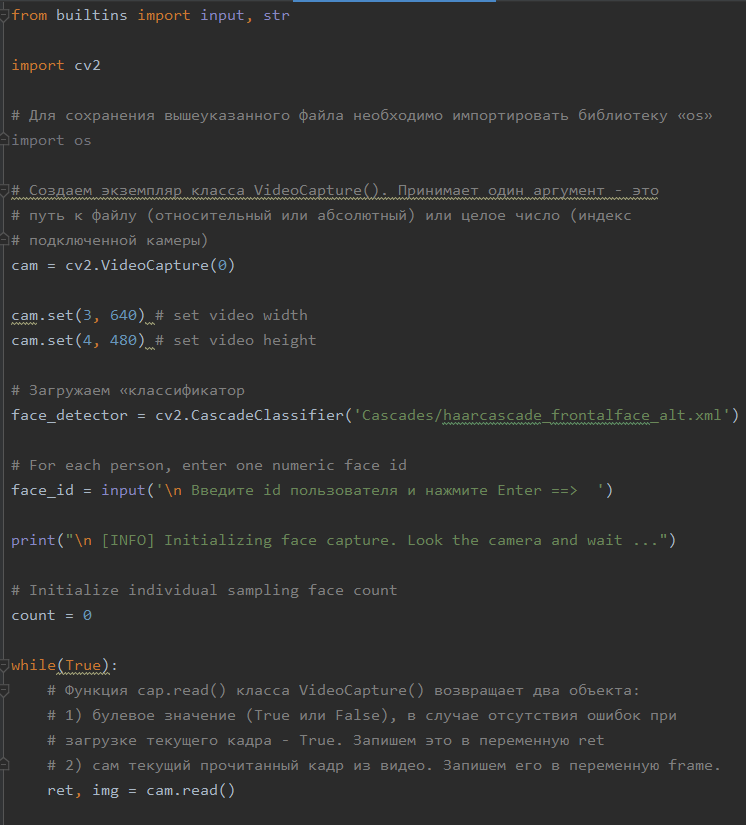
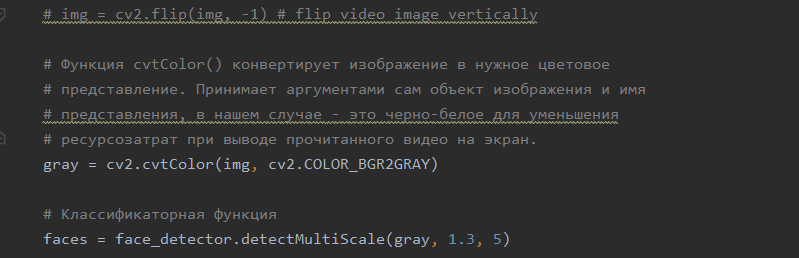


Рисунок 3 – Определение лица

## Сбор данных

То, что мы будем делать здесь, начинается с последнего шага (Определение Лица), мы просто создадим набор данных, где мы будем хранить для каждого идентификатора, группу фотографий в сером цвете с той частью, которая использовалась для обнаружения лица.





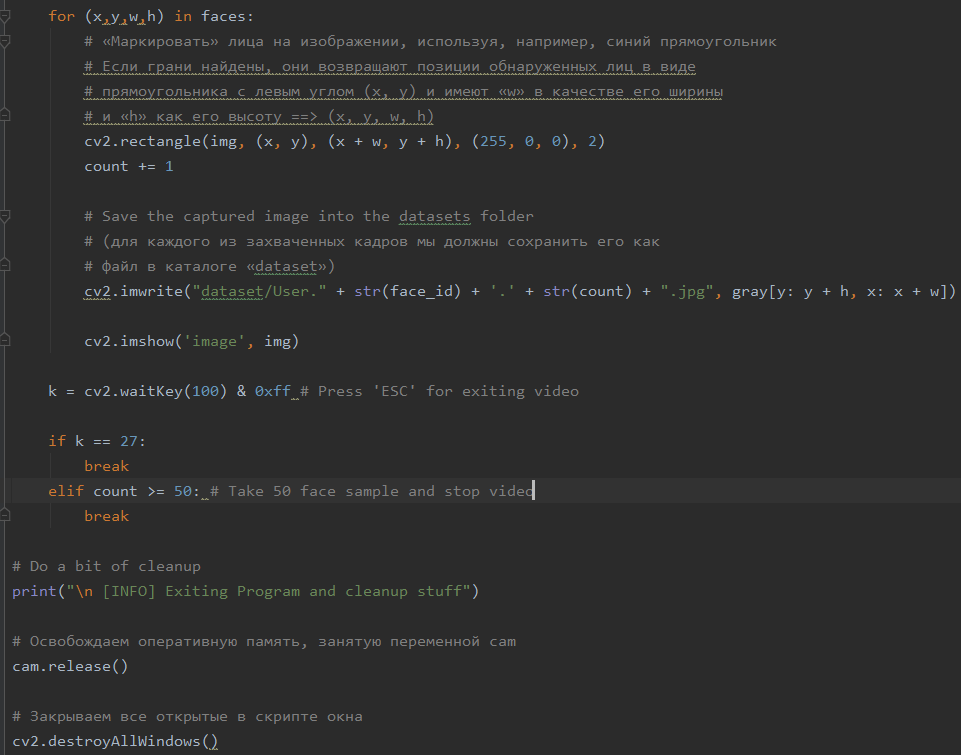


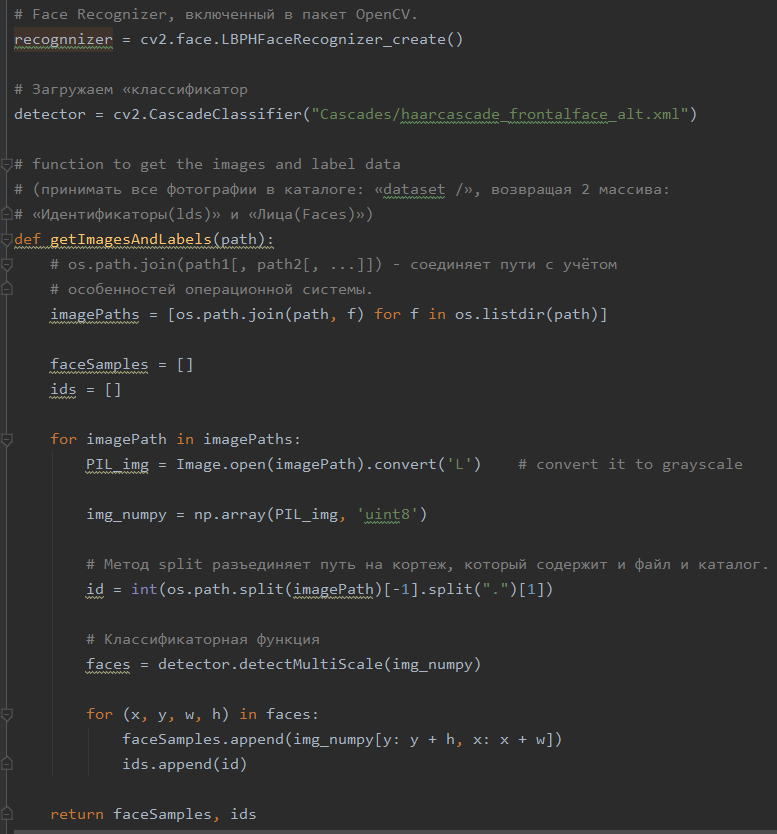
Рисунок 4 – Сбор данных

Код очень похож на код, который мы писали для обнаружения лиц. Мы добавили, что это была «команда ввода» для захвата идентификатора пользователя, который должен быть целым числом (1, 2, 3 и т. д.). для каждого из захваченных кадров мы должны сохранить его как файл в каталоге «dataset».

## Тренировка

На этом этапе мы должны взять все пользовательские данные из нашего набора данных и «инструктор» в OpenCV Recognizer. Это делается непосредственно с помощью определенной функции OpenCV. Результатом будет файл .yml, который будет сохранен в каталоге «trainer /».





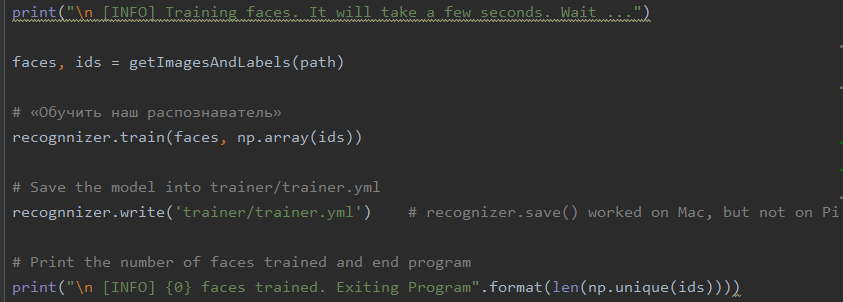


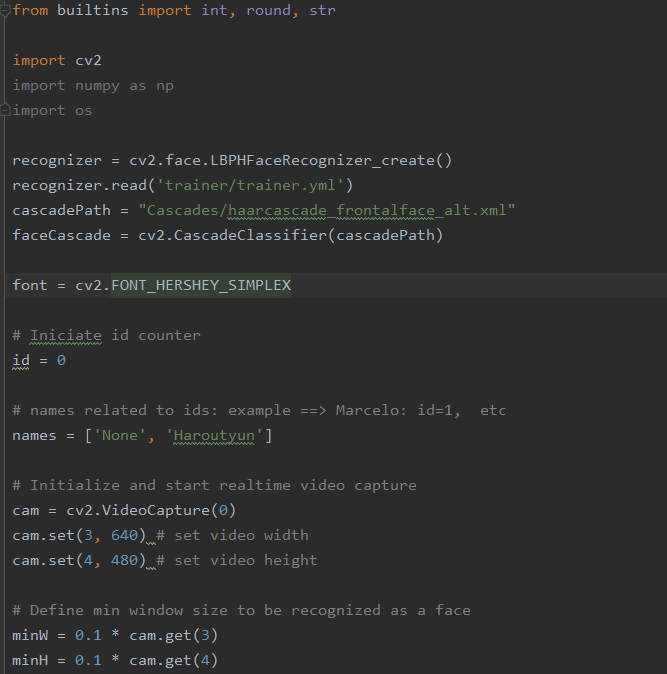
Рисунок 5 – Тренировка

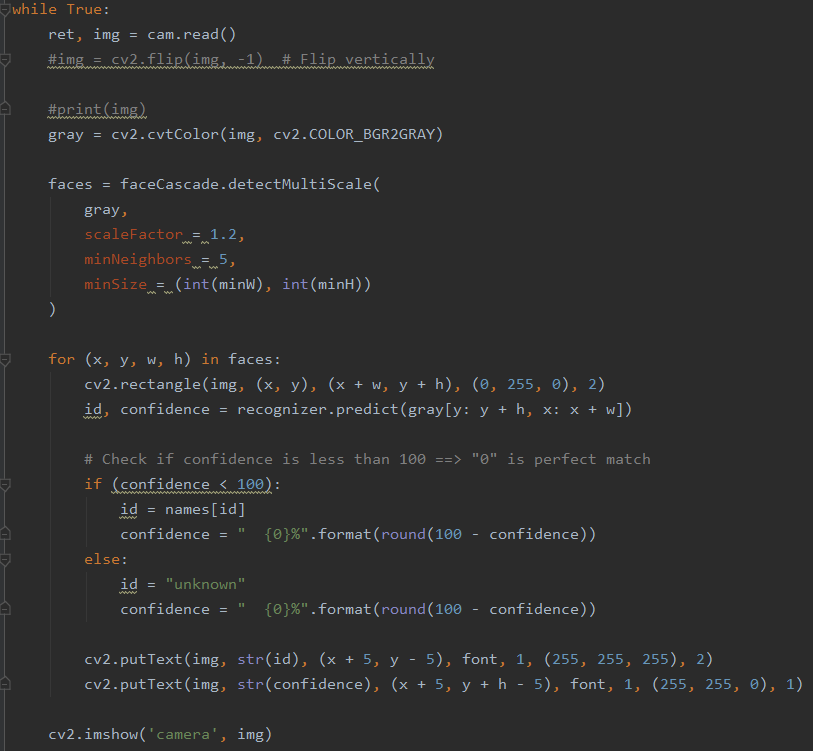
Мы использовали в качестве распознавателя, LBPH (LOCAL BINARY PATTERNS HISTOGRAMS) Face Recognizer, включенный в пакет OpenCV.

В результате файл с именем «trainer.yml» будет сохранен в каталоге тренера, который был ранее создан нами.

## Распознавание лица

Теперь мы достигли заключительной фазы нашего проекта. Здесь мы увидим лицо на нашей камере, и, если бы этот человек был сфотографирован и обучен раньше, наш распознаватель сделает «обработку», возвращающую свой идентификатор и индекс, показавший, насколько уверен в этом признаке распознаватель.





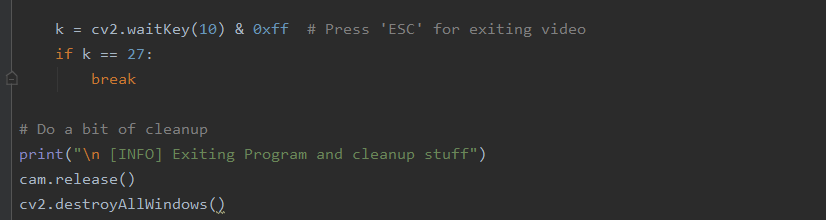


Рисунок 6 – Распознавание лица

Если распознаватель может определить лицо, мы помещаем текст над изображением с вероятным идентификатором и насколько «вероятность (probability)» в% соответствует правилу («вероятность» = 100 - индекс доверия). Если нет, на лицо помещается ярлык «неизвестный».

## Заключение

В данной работе мы использовали камеру компьютера для распознавания лица в режиме реального времени с помощью Python и OpenCV.

Программа обнаруживает лицо и собирает данные. Далее происходит обучение распознавателя и в конце концов – распознавание лица.

Всё работает хорошо.