

VI Международная конференция учащихся  
НАУЧНО-ТВОРЧЕСКИЙ ФОРУМ

Секция IT-технологии и современность

**«Компьютерное зрение на службе у  
человека»**

**Автор:**

Никульченков Георгий Александрович,  
ученик 11 «Б» класса МАОУ СОШ №7  
города Когалыма

**Руководитель:**

Полукарикова Алла Сергеевна,  
учитель информатики МАОУ СОШ №7  
города Когалыма

2025г.

## Оглавление

Введение .....	2
Основная часть.....	3
1. Общие сведения о компьютерном зрении .....	3
2. Этапы создания/разработки .....	4
2.1 Структура программы.....	5
2.2 Среда разработки.....	5
2.3 Этапы создания/разработки .....	6
2.4 Апробация.....	8
Заключение.....	9
Список источников.....	10
Приложение.....	11

## Введение

ИТ- технологии для меня вызывают приоритетный интерес среди множества инновационных направлений в информатике. Именно этим направлением в проектной деятельности мне довелось заниматься еще в средней школе. В старшей школе меня заинтересовало, как можно реализовать умение программировать при использовании технических устройств.

На занятиях кружка в школе мы участвовали в уроках Национальной технологической олимпиады<sup>1</sup> (НТО), где просматривали профиль "Компьютерное зрение", преподаватель предложила по желанию познакомиться с ней ближе. Тема показалась мне достаточно интересной, актуальной и востребованной в наше время. Компьютерное зрение является частью машинного обучения и искусственного интеллекта.

**Актуальность проекта** заключается в том, что Компьютерное зрение, как технология, является составляющей современных интеллектуальных автоматизированных систем, служит инструментом развития технологии нейросетей, которые используются во множестве сфер человеческой жизни. Другими словами, системы компьютерного зрения используются для глубокого обучения нейросетей.

Чтобы в деталях на практике ознакомиться с данной технологией, было решено выбрать следующую **цель проекта**:

*создание приложения, которое использует технологию компьютерного зрения для распознавания лиц.*

### **Задачи проекта:**

1. Сбор необходимого теоретического материала (какие технологии используются и область их применения);
2. Выбор среды и языка программирования для создания программы;
3. Написание программы;
4. Тестирование готового программного продукта.

### **Практическая значимость проекта.**

Программа, которая создается при разработке проекта является общей для технологии компьютерного зрения по распознаванию лиц. Распознавание лиц применяют в системах безопасности различного уровня. Можно внедрить ее в пределах школы на входе, как альтернативу пропускной системы в образовательную организацию, так как это усовершенствованный тип обнаружения объектов, который не только распознает человеческое лицо на изображении, но и идентифицирует конкретного человека.

---

<sup>1</sup> <https://nti-lesson.ru/> сайт Урок НТО

## Основная часть

### 1. Общие сведения о компьютерном зрении

Для начала разберёмся, что же такое компьютерное зрение? Компьютерное зрение относится к технологии создания искусственных систем путём глубокого обучения — разновидности машинного обучения, которое использует для решения задач нейронные сети. Данные системы получают информацию из изображений, производя обнаружение, отслеживание и классификацию объектов. Видеоданные могут быть представлены множеством форм [1].

Миссия компьютерного зрения — научить вычислительную машину видеть и понимать окружение с помощью цифровых фотографий и видеозаписей. Для достижения этой цели используются три компонента:

- получение изображений;
- обработка информации;
- анализ данных.

Получение изображений, это процесс превращения аналогового мира в цифровой вид. Для этого могут использоваться различные инструменты, такие как: веб-камеры, цифровые и зеркальные фотоаппараты и т.д.

Полученные данные необходимо обработать и проанализировать. Для этого используют низкоуровневую обработку данных. Она необходима для определения краев, точек и сегментов изображения, являющихся простыми геометрическими фигурами.

Как правило, обработка данных осуществляется с помощью сложных математических алгоритмов. Известными методами низкоуровневого анализа являются:

- выделение границ (edge detection);
- сегментация;
- классификация и обнаружение объектов.

Анализ и понимание изображений, это последний шаг в компьютерном зрении, позволяющий машинам принимать собственные решения. На этом этапе используются высокоуровневые данные, полученные из предыдущего шага. Примером высокоуровневого анализа может быть отображение трехмерной сцены, распознавание или отслеживание объектов [2].

На сегодня методы компьютерного зрения используются во многих областях, таких как:

- Безопасность
- Распознавание лиц

- Беспилотные автомобили
- Робототехника
- Дополненная реальность
- Распознавание движений и жестов
- Восстановление и обработка изображений

Проведя анализ областей применения компьютерного зрения, я решил остановиться конкретно на "распознавании лиц". Поэтому мой будущий продукт будет связан именно с этой областью. Теперь узнаем про неё подробнее.

Что же такое "распознавание лиц"? Это способ идентификации или подтверждения личности человека по его лицу. Систему распознавания лиц можно использовать для идентификации людей на фотографиях, видео или в режиме реального времени [3].

Технологии распознавания лиц применяются в самых разнообразных сферах [4]:

- Обеспечение безопасности в местах большого скопления людей;
- Системы охраны, избежание незаконного проникновения на территорию объекта, поиск злоумышленников;
- Фейс-контроль в сегменте общепита и развлечений, поиск подозрительных и потенциально опасных посетителей;
- Верификация банковских карт;
- Онлайн-платежи;
- Контекстная реклама, цифровой маркетинг, Intelligent Signage и Digital Signage;
- Фототехника;
- Криминалистика;
- Телеконференции;
- Мобильные приложения;
- Поиск фото в больших базах фотоснимков;
- Отметка людей на фото в социальных сетях и многие другие.

## **2. Этапы создания/разработки**

Выбрав область применения компьютерного зрения, а именно распознавание лиц, я начал искать и изучать весь необходимый материал в сети Интернет. Это был достаточно длительный процесс, о чем свидетельствует список использованных сайтов. Изучив всю необходимую информацию, началось создание структуры программы и выбор среды для её разработки.

## 2.1 Структура программы

Компьютерное зрение работает в три основных этапа, что определяет структуру программы:

- Получение изображения (изображения, даже большие, можно получать в режиме реального времени с помощью видео, фотографий или 3D-технологий для анализа).
- Обработка изображения (модели глубокого обучения автоматизируют большую часть этого процесса, но зачастую модели обучаются, получая сначала тысячи маркированных или заранее идентифицированных изображений).
- Понимание изображения (этап интерпретации, на котором объект идентифицируется или классифицируется).

## 2.2 Среда разработки

Определившись со структурой проекта, пришло время выбрать среду и язык программирования для разработки программы по распознаванию лиц.

Для написания программ для распознавания лиц, используется библиотека алгоритмов компьютерного зрения и обработки изображений - OpenCV,. Данная библиотека поддерживается языками C/C++, Python, R, Java, Ruby, Matlab, Lua и другими [5].

Выбирая необходимый язык, я ориентировался на свои уже имеющиеся знания в программировании. Кроме того, я обращал внимание на лёгкость в освоении и применении того или иного языка программирования, а также на наличие подробной документации в свободном доступе.

После тщательного анализа и сравнения я остановился на языке программирования Python, который является одним из самых популярных выборов для программистов в области машинного обучения. У Python есть ряд особенностей, которые делают его отлично подходящим для программирования нейросетей — этот язык легко изучать и читать. Названия функций и сам код написаны на простом английском языке [6]. К тому же, у меня уже были базовые знания данного языка программирования.

В свободном доступе находится множество статей [7], видео уроков [8] и документаций [9] по использованию библиотеки OpenCV на Python, что также соответствует одному из вышеперечисленных критериев.

В качестве среды для написания программы я решил выбрать Visual Studio Code, так как он поддерживает большое количество языков программирования, включая Python, и

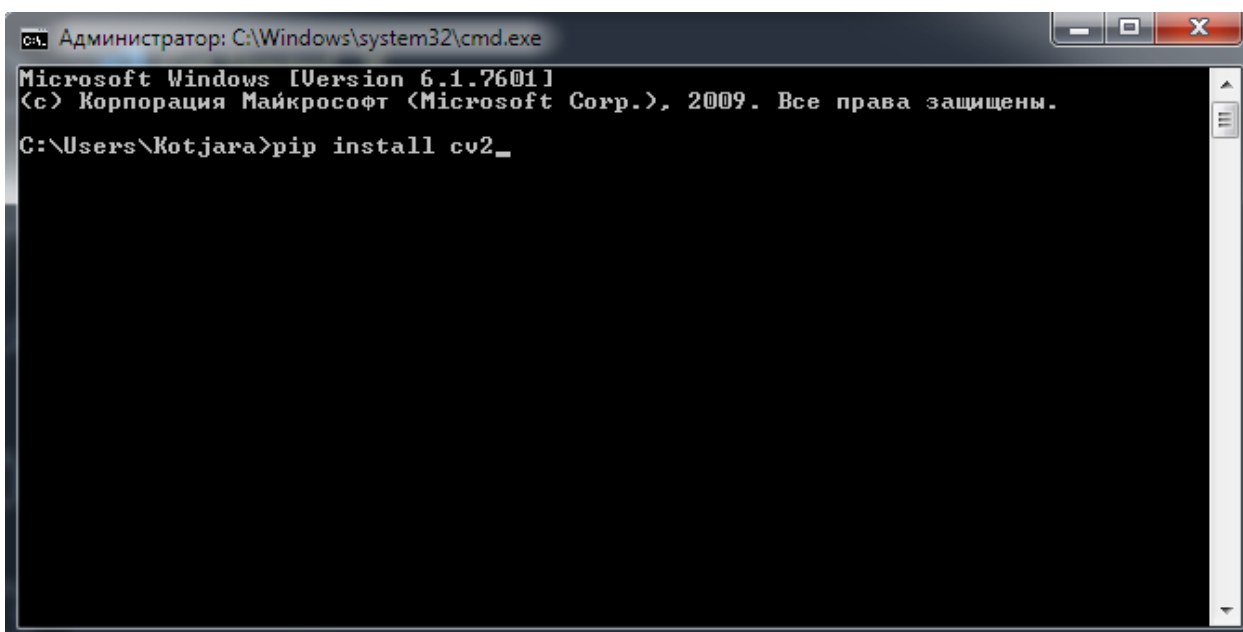
находится в свободном доступе. К тому же, там есть возможность для установки различных расширений и других вспомогательных инструментов для упрощения написания кода программы. Также у меня уже имеется опыт работы с данной средой разработки ещё со средней школы.

Именно на языке программирования Python, с использованием Visual Studio Code, я и буду создавать итоговый продукт.

### 2.3 Этапы создания/разработки

Определившись со структурой, средой и языком программирования, осталось лишь написать код для будущего продукта.

Для этого необходимо установить необходимую библиотеку OpenCV на Python через командную строку CMD. Для этого мы используем специальную команду: `pip install cv2` (рис. 1)



```
Администратор: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\Kotjara>pip install cv2_
```

Рис. 1

После установки библиотеки следует установка всех необходимых файлов для работы с данной библиотекой [10], [11]. В данных файлах находятся все необходимые математические формулы и алгоритмы распознавания лиц для уже обученной нейросети.

Все установленные файлы были перемещены в заранее установленную папку. Именно в ней и будет создан файл для будущей программы.

Следующим шагом идёт открытие папки через Visual Studio Code. Для этого требуется зайти в программу и нажать на кнопку "Open Folder" (рис. 2)

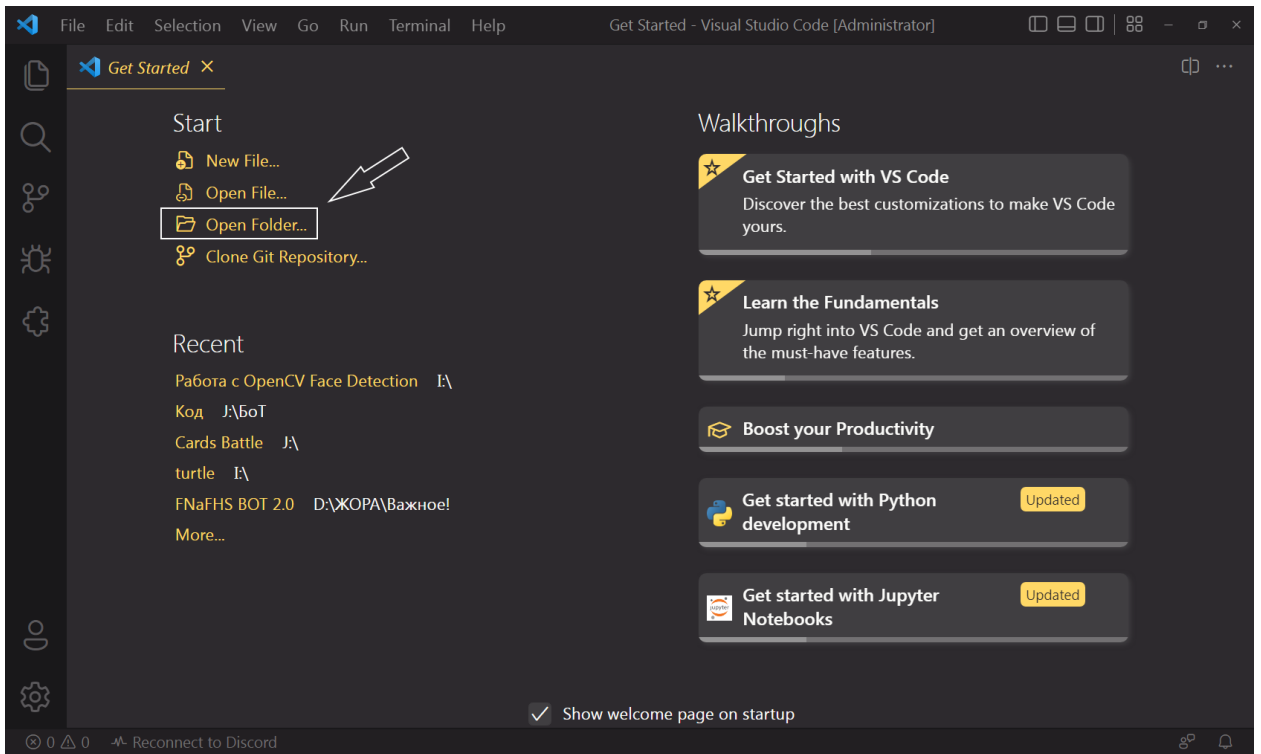


Рис. 2

После выбора необходимой папки, при помощи боковой панели, создаётся файл будущей программы с необходимым названием и расширением (рис. 3)

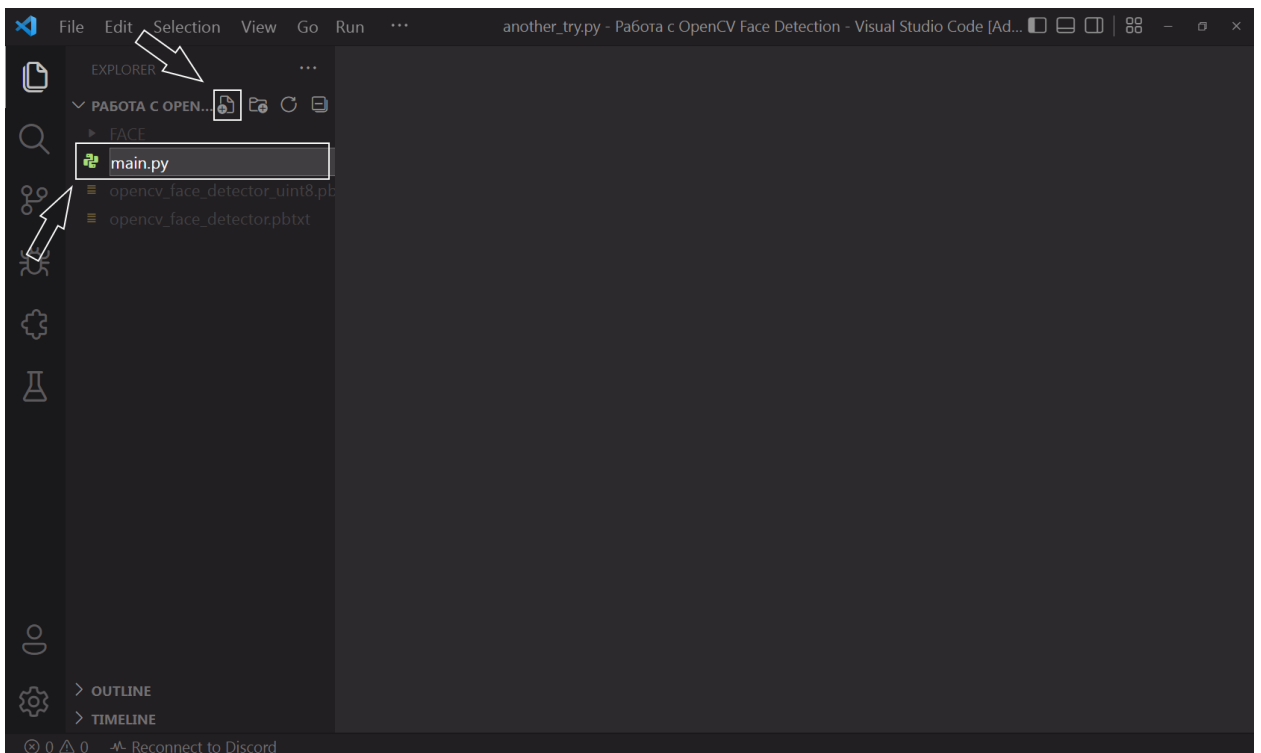


Рис. 3



Самый трудоёмкий момент разработки проекта, это конечно, прописывание программного кода. Именно он позволит распознать лица при помощи нейросети через подключённое к компьютеру видеоустройство.

## 2.4 Апробация

Последним шагом моего проекта стала апробация программы при помощи веб-камеры на моём компьютере (рис. 4)

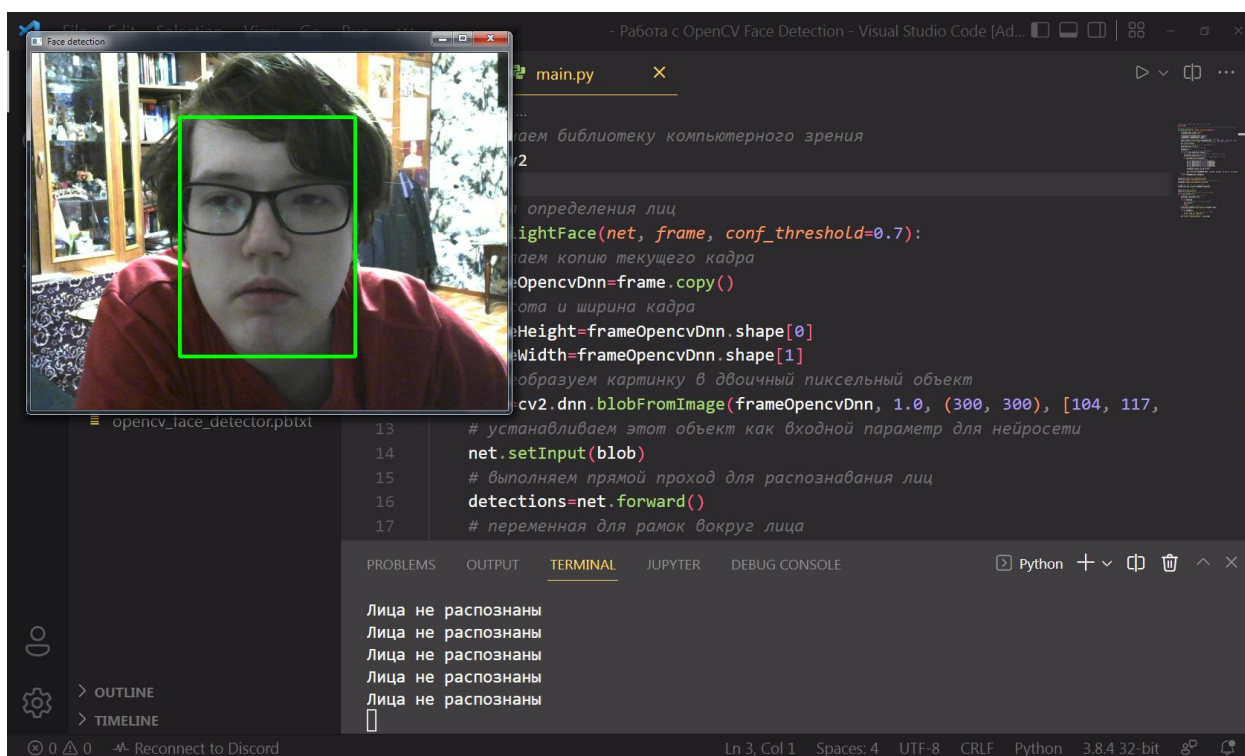


Рис. 4

Это помогло мне проверить работоспособность программы и выявить ошибки, и недочёты, чтобы в дальнейшем их исправить.

## **Заключение**

Работа над прикладным проектом – это всегда творческий процесс, в результате которого мы приобретаем ценный опыт самостоятельной учебной деятельности: работы с информацией и ее источниками; учимся анализировать и отбирать необходимую информацию.

Приобретение навыков в ходе самообразования помогает на последующем образовательном пути.

Работа над созданием программы распознавания лиц позволила мне почувствовать себя программистом, причастным к современным информационным технологиям. Это заставило меня осознать, что работа с данными и информацией мне действительно нравится, и я хочу посвятить этому свое будущее.

Кроме того, при дальнейшем развитии этого продукта имеется возможность использовать компьютерное зрение на входе в школу, а также в школьных коридорах и классах для определения дисциплинарных нарушений со стороны учащихся.

Каждый современный пользователь компьютера сталкивается с приложениями, где используется компьютерное зрение, например, системы авторизации и контроля доступа. Если вы самостоятельно можете «научить компьютер видеть и понимать картинку», это значит иметь надежного помощника. Компьютер на основе алгоритмов и камеры станет надежной противоугонной системой для личного автомобиля, о которой сложно догадаться вандалу или правонарушителю. При желании есть возможность создания подобной программы, что доказано на примере моего проекта.

Чувствую удовлетворенность результатом своей работы. Мне удалось выполнить все задачи, которые я перед собой ставил.

## Список источников

1. <https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5%D0%B7%D1%80%D0%B5%D0%BD%D0%B8%D0%B5>
2. <https://forklog.com/cryptorium/ai/что-такое-компьютерное-зрение-машинное-обучение>
3. <https://www.kaspersky.ru/resource-center/definitions/what-is-facial-recognition>
4. [https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B%D1%80%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%D0%BB%D0%B8%D1%86 \(Facial recognition\)](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B%D1%80%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%D0%BB%D0%B8%D1%86%20(Facial%20recognition))
5. <https://ru.wikipedia.org/wiki/OpenCV>
6. <https://kata.academy/article/5-yazykov-programmirovaniya-II-dlya-nachinayushchikh>
7. <https://itproger.com/course/opencv>
8. [https://www.youtube.com/playlist?list=PL0IO\\_mIqDDFUAQ2RdAgLp6Tj\\_fREcxk6T](https://www.youtube.com/playlist?list=PL0IO_mIqDDFUAQ2RdAgLp6Tj_fREcxk6T)
9. [https://docs.opencv.org/4.x/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html)
10. <https://disk.yandex.ru/d/FRJmkIGtci-SbQ>
11. <https://disk.yandex.ru/d/QQTnul-KNG2kXg>

# Приложение

(Программный код продукта)

```
# подключаем библиотеку компьютерного зрения
import cv2

# функция определения лиц
def highlightFace(net, frame, conf_threshold=0.7):
    # делаем копию текущего кадра
    frameOpencvDnn=frame.copy()
    # высота и ширина кадра
    frameHeight=frameOpencvDnn.shape[0]
    frameWidth=frameOpencvDnn.shape[1]
    # преобразуем картинку в двоичный пиксельный объект
    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)
    # устанавливаем этот объект как входной параметр для нейросети
    net.setInput(blob)
    # выполняем прямой проход для распознавания лиц
    detections=net.forward()
    # переменная для рамок вокруг лица
    faceBoxes=[]
    # перебираем все блоки после распознавания
    for i in range(detections.shape[2]):
        # получаем результат вычислений для очередного элемента
        confidence=detections[0,0,i,2]
        # если результат превышает порог срабатывания — это лицо
        if confidence>conf_threshold:
            # формируем координаты рамки
            x1=int(detections[0,0,i,3]*frameWidth)
            y1=int(detections[0,0,i,4]*frameHeight)
            x2=int(detections[0,0,i,5]*frameWidth)
            y2=int(detections[0,0,i,6]*frameHeight)
            # добавляем их в общую переменную
            faceBoxes.append([x1,y1,x2,y2])
```

```

# рисуем рамку на кадре
cv2.rectangle(frameOpenCvDnn, (x1,y1), (x2,y2), (0,255,0), int(round(frameHeight/150)), 8)
# возвращаем кадр с рамками
return frameOpenCvDnn,faceBoxes

# загружаем веса для распознавания лиц
faceProto="opencv_face_detector.pbtxt"
# и конфигурацию самой нейросети — слои и связи нейронов
faceModel="opencv_face_detector_uint8.pb"

# запускаем нейросеть по распознаванию лиц
faceNet=cv2.dnn.readNet(faceModel,faceProto)

# получаем видео с камеры
video=cv2.VideoCapture(0)
# пока не нажата любая клавиша — выполняем цикл
while cv2.waitKey(1)<0:
    # получаем очередной кадр с камеры
    hasFrame,frame=video.read()
    # если кадра нет
    if not hasFrame:
        # останавливаемся и выходим из цикла
        cv2.waitKey()
        break
    # распознаём лица в кадре
    resultImg,faceBoxes=highlightFace(faceNet,frame)
    # если лиц нет
    if not faceBoxes:
        # выводим в консоли, что лицо не найдено
        print("Лица не распознаны")
    # выводим картинку с камеры
    cv2.imshow("Face detection", resultImg)

```