МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОУ НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. Р. Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА «ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

Лабораторная работа 3

по дисциплине:

“Моделирование информационных процессов и систем”

На тему:

«Верификация человека по изображению лица»

Выполнил

Студент группы:

М19-ИВТ-3

Суворкин П.А.

Проверил:

Гай В.Е.

Н.Новгород

2020 г.

**Задание:** используя библиотеку DeepFace реализовать систему верификации человека по изображению лица. Сделать свою фотографию, выполнить детектирование лица на изображении, сформировать вектор признаков лица, используя указанные в варианте задания детектор и модель. Захватить видеопоток с веб-камеры, выполнить верификацию в реальном масштабе времени. Определить устойчивость указанного в варианте задания модели и детектора к повороту и наклону головы.

**Вариант:** 33

**Face Detector:** 3. Dlib

**Face recognition model:** 3. OpenFace

Deepface-это легкий фреймворк для распознавания лиц и анализа атрибутов лица (возраст, пол, эмоции и раса) для python. Это гибридный фреймворк распознавания лиц, включающий в себя самые современные модели: VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace и Dlib. Библиотека в основном основана на Keras и TensorFlow.

Современный конвейер распознавания лиц состоит из 4 общих этапов:

1. обнаружение,
2. выравнивание,
3. представление
4. проверка.

Deepface обрабатывает все эти общие этапы в фоновом режиме.

Функция верификации под интерфейсом deepface предлагает проверять пары лиц одного человека или разных.

Распознавание лиц требует применения верификации лиц несколько раз. Здесь deepface предлагает готовую функцию поиска для обработки этого действия. Она хранит представления базы данных лиц, потому нет необходимости искать ее снова. Таким образом, также можно применить набор данных распознавания лиц. Функция find возвращает фрейм данных pandas, если передается один путь изображения, а также возвращает список фреймов данных pandas, если передается список путей изображения.

Deepface-это гибридный пакет распознавания лиц. В настоящее время он включает в себя самые современные модели распознавания лиц: VGG-Face , Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace и Dlib. Конфигурация по умолчанию проверяет лица с помощью модели VGG-Face. Можно установить базовую модель во время проверки, как показано ниже.

models = ["VGG-Face", "Facenet", "OpenFace", "DeepFace", "DeepID", "ArcFace", "Dlib"]

for model in models:

result = DeepFace.verify("img1.jpg", "img2.jpg", model\_name = model)

df = DeepFace.find(img\_path = "img1.jpg", db\_path = "C:/workspace/my\_db", model\_name = model)

FaceNet, VGG-Face, ArcFace и Dlib превосходят OpenFace, DeepFace и DeepID по результатам экспериментов. Кроме того, FaceNet получил 99,65%; ArcFace получил 99,40%; Dlib получил 99,38%; VGG-Face получил 98,78%; OpenFace получил 93,80% баллов точности по набору данных LFW, в то время как люди могут показать результат только 97,53%.

**Сходство**

Модели распознавания лиц - это обычные сверточные нейронные сети, и они отвечают за представление лиц в виде векторов. Решение о верификации основывается на расстоянии между векторами. Мы можем классифицировать пары, если их расстояние меньше порога.

Расстояние может быть найдено с помощью различных метрик, таких как косинусное сходство, Евклидово расстояние и форма L2. Конфигурация по умолчанию находит косинусное сходство.

**Анализ Атрибутов Лица**

Deepface также предлагает анализ атрибутов лица, включая возраст, пол, выражение лица:

* гнев,
* страх,
* нейтральная эмоция,
* печаль,
* отвращение,
* счастье
* удивление

и расу (включая азиатов, белых, арабов, индийцев, латиноамериканцев и чернокожих).

[](https://raw.githubusercontent.com/serengil/deepface/master/icon/stock-2.jpg)

Здесь возрастная модель получила ± 4,65 MAE; гендерная модель получила точность 97,44%, точность 96,29% и отзыв 95,05%.

**Потоковая передача и анализ в реальном времени**

deep face также возможно использовать для видео в реальном времени.

Вызов функции stream под интерфейсом DeepFace позволит получить доступ к веб-камере и применить как распознавание лиц, так и анализ атрибутов лица. Потоковая функция ожидает папку базы данных, включающую изображения лиц. VGG-Face - это модель распознавания лиц по умолчанию, а косинусное сходство - это метрика расстояния по умолчанию, аналогичная функции verify. Функция начинает анализировать, может ли она последовательно фокусировать лицо на 5 кадрах. Затем она показывает результат через 5 секунд.

Несмотря на то, что распознавание лиц основано на однократном обучении, также можно использовать несколько изображений лица человека. Для этого следует изменить структуру каталогов, как показано ниже.

user

├── database

│ ├── Alice

│ │ ├── Alice1.jpg

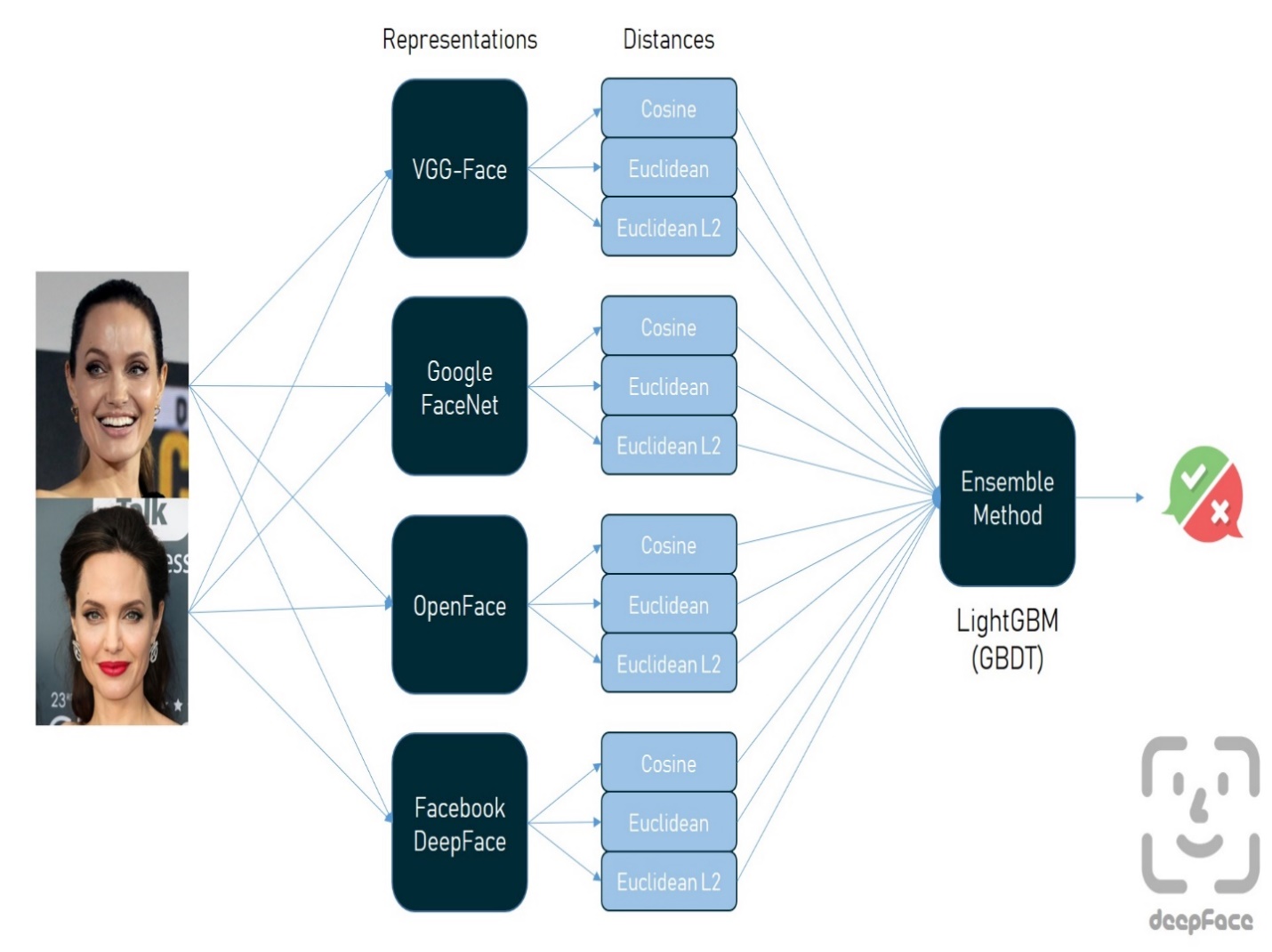
│ │ ├── Alice2.jpg

│ ├── Bob

│ │ ├── Bob.jpg

**Ансамблевое обучение для распознавания лиц**

Задача распознавания лиц может быть обработана несколькими моделями и метриками сходства. Здесь deepface предлагает специальное повышающее и комбинированное решение для повышения точности задачи распознавания лиц. Это обеспечивает огромное улучшение показателей точности. Люди могли бы иметь 97,53% баллов для задач распознавания лиц, в то время как этот ансамблевый метод проходит точность человеческого уровня и получает точность 98,57%. С другой стороны, это работает гораздо медленнее, чем отдельные модели.

[](https://raw.githubusercontent.com/serengil/deepface/master/icon/stock-4.jpg)

**Детекторы Лиц**

Распознавание и выравнивание лиц - это ранние этапы современного конвейера распознавания лиц. OpenCV haar cascade, SSD, Dlib и методы сопоставления завернуты в deep face в качестве детектора. Можно дополнительно передать пользовательский детектор функциям в интерфейсе deepface. MTCN-это детектор по умолчанию.

backends = ['opencv', 'ssd', 'dlib', 'mtcnn']

for backend in backends:

#face detection and alignment

detected\_face = DeepFace.detectFace("img.jpg", detector\_backend = backend)

#face verification

obj = DeepFace.verify("img1.jpg", "img2.jpg", detector\_backend = backend)

#face recognition

df = DeepFace.find(img\_path = "img.jpg", db\_path = "my\_db", detector\_backend = backend)

#facial analysis

demography = DeepFace.analyze("img4.jpg", detector\_backend = backend)

MTCNN, похоже, показывает лучшие результаты на этапах обнаружения и выравнивания, но он медленнее, чем SSD.

**Результат работы:**

Для тестирования были выбраны 4 фотографии:

1 в анфас

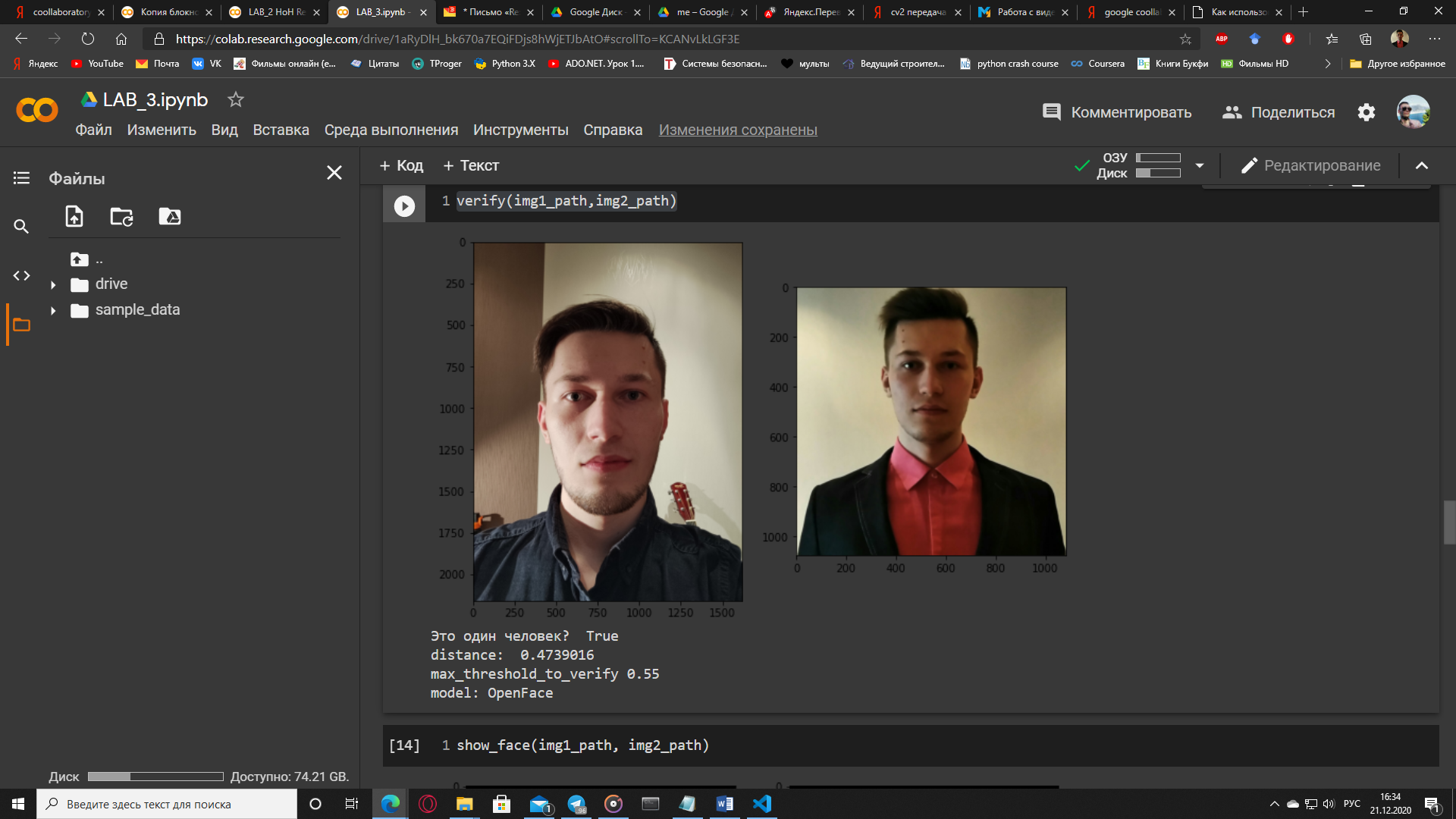
2 в анфас

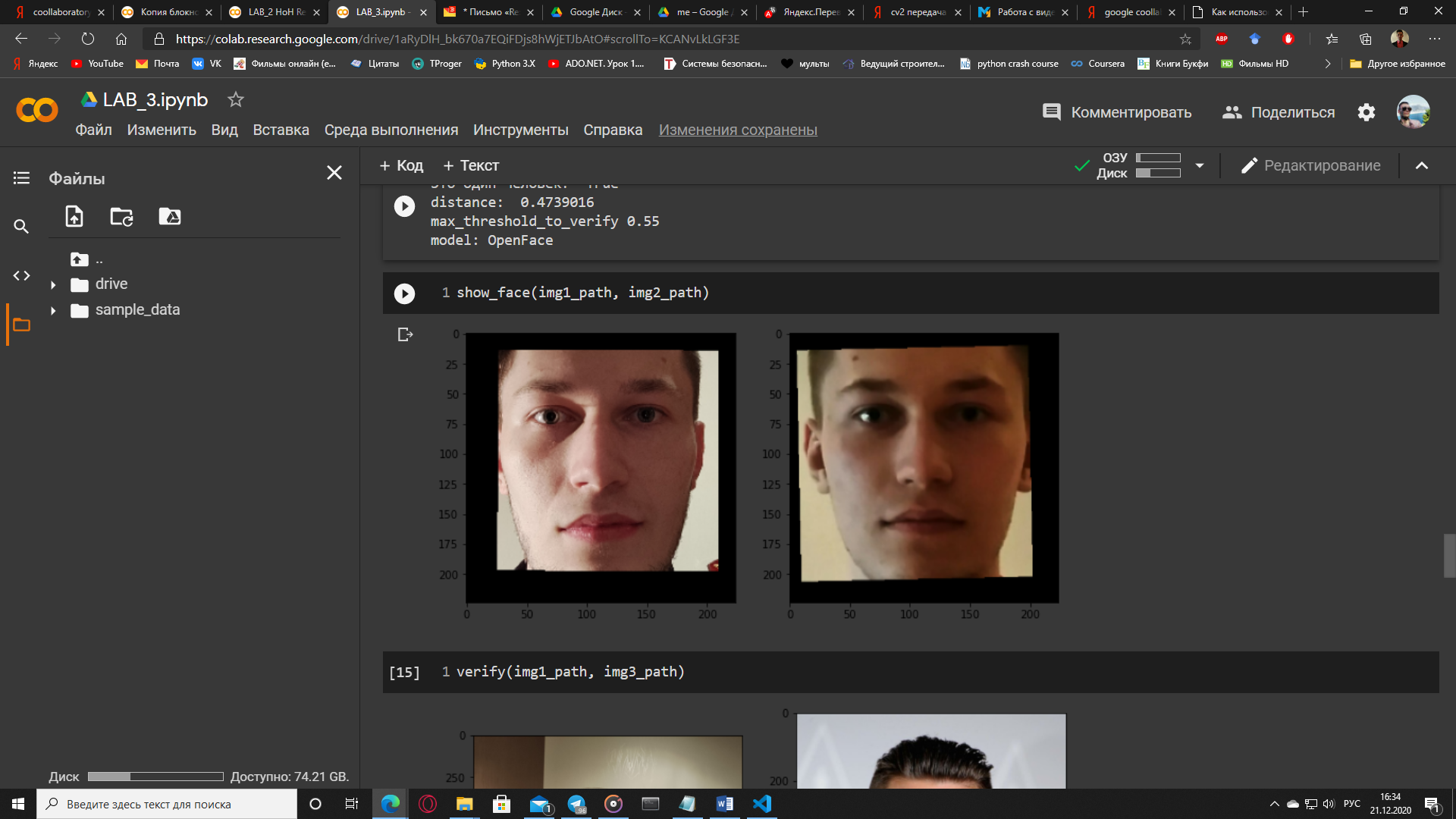
3 другой человек в анфас

4 в профиль



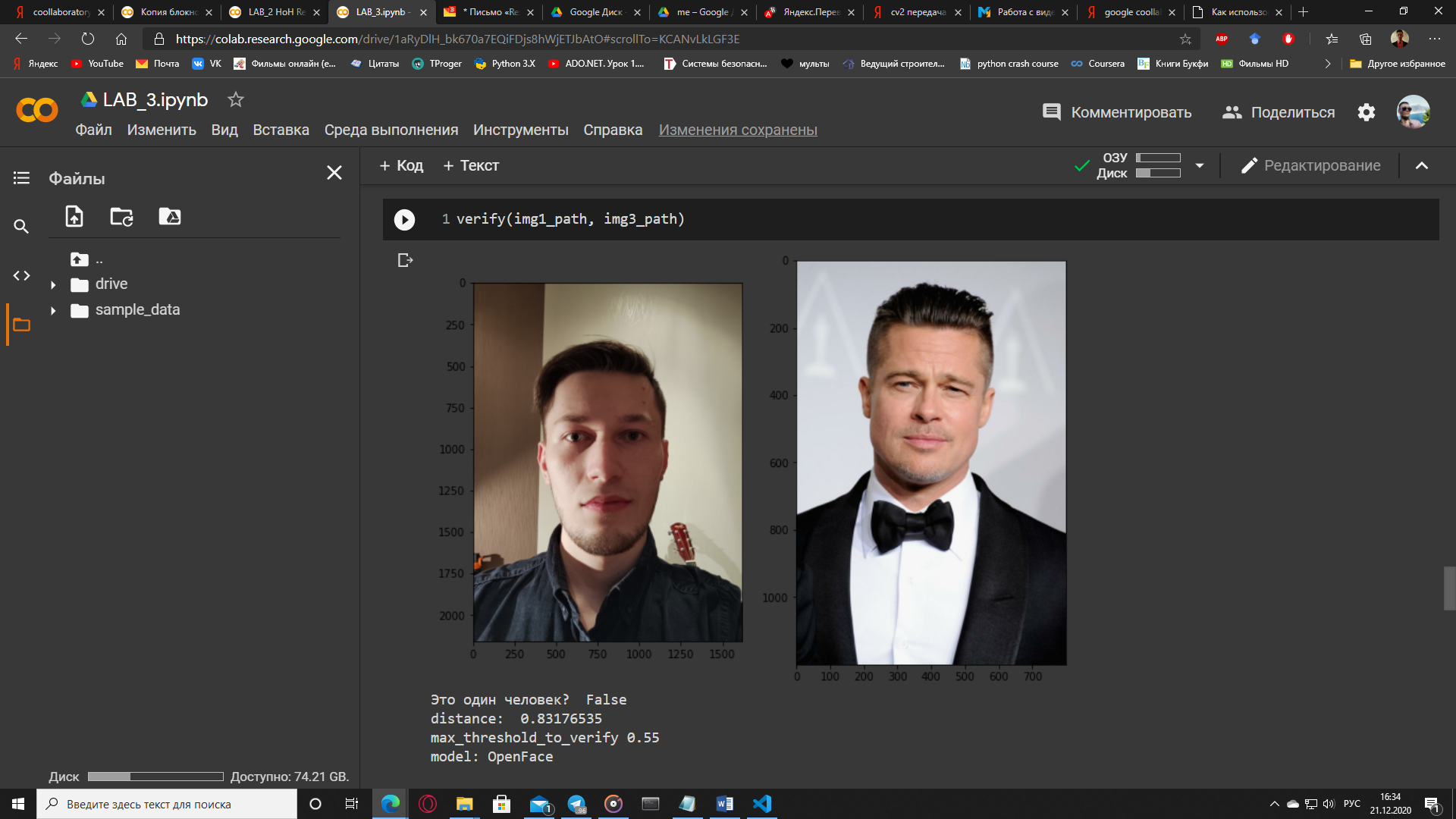
**1 эксперимент:** я взял две свои фотографии, сделанные в разное время. На фото хорошо видно лицо.

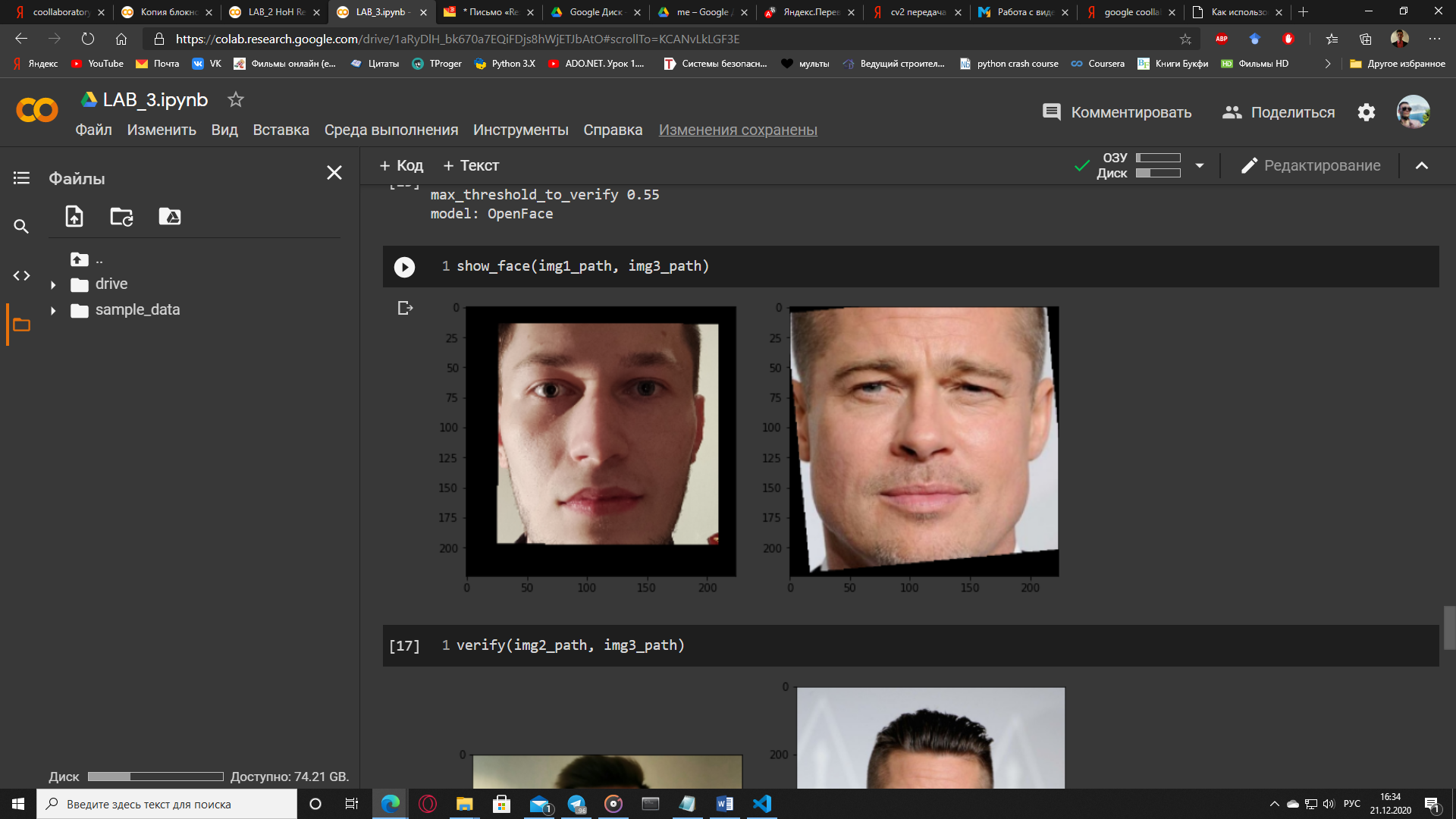




**Результат:** сеть выдала совпадение. Евклидово расстояние 0.47. Чем больше похожи люди на фото, тем оно меньше. Порог по умолчанию стоит 0.55 . Сеть справилась успешно.

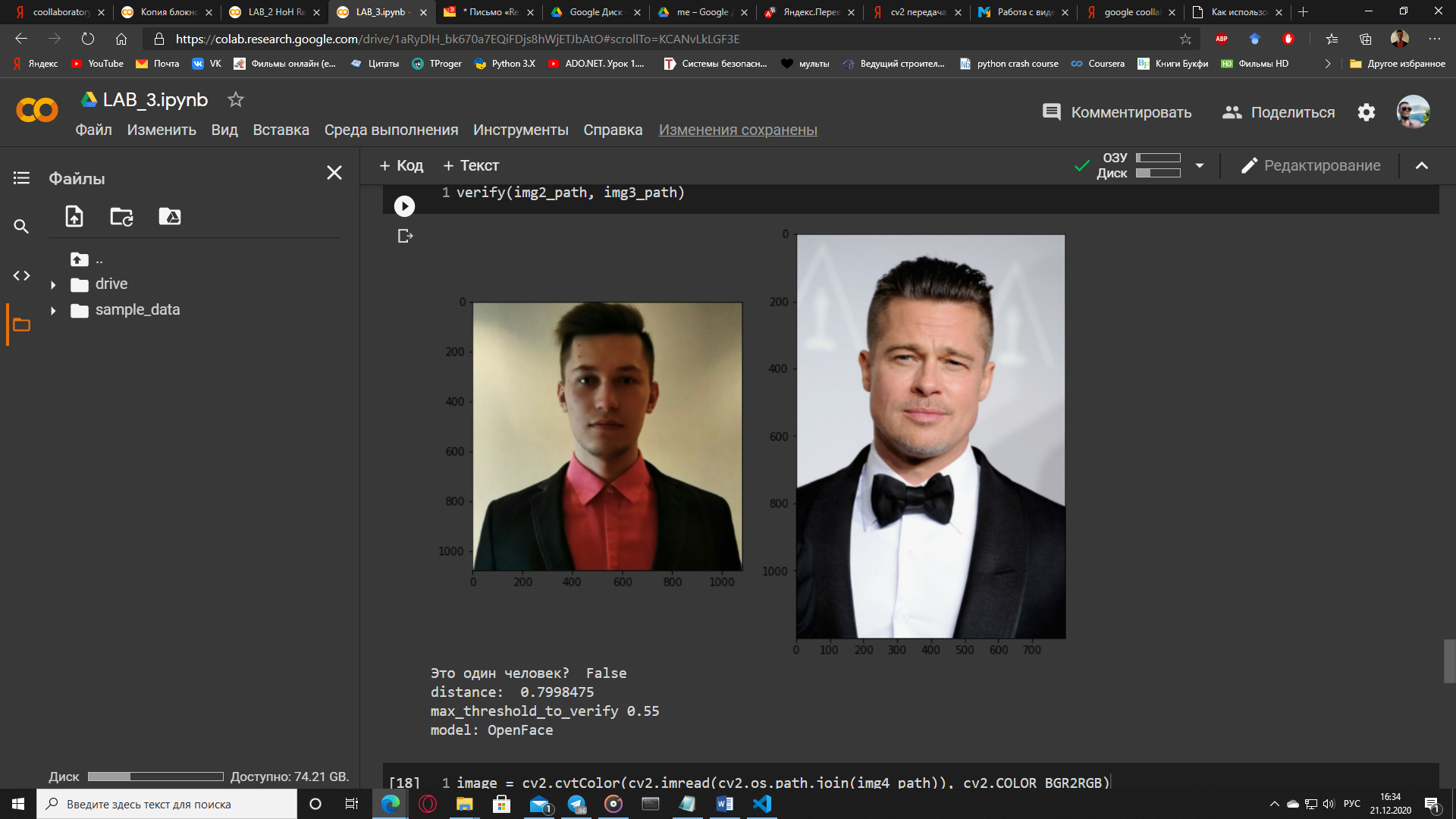
**2 эксперимент:** я взял свою фотографию и фото Брэд Питта. На фото хорошо видно лицо.

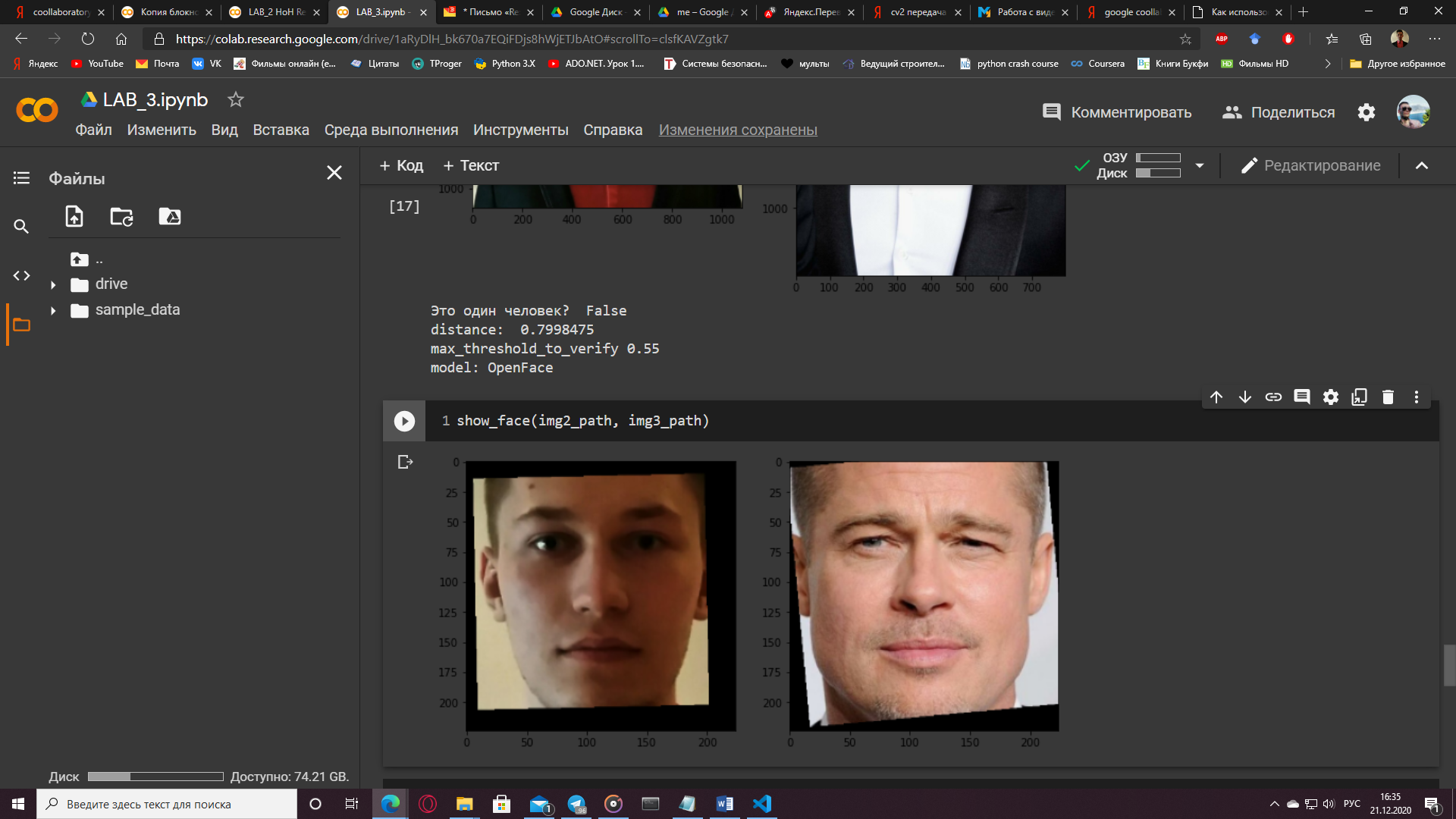




**Результат:** сеть выдала не совпадение. Евклидово расстояние 0.83. Порог по умолчанию стоит 0.55. Разница очень велика, потому что представлены 2 разных человека. Сеть справилась успешно.

**3 эксперимент:** я взял другую свою фотографию и фото Брэд Питта. На фото хорошо видно лицо.



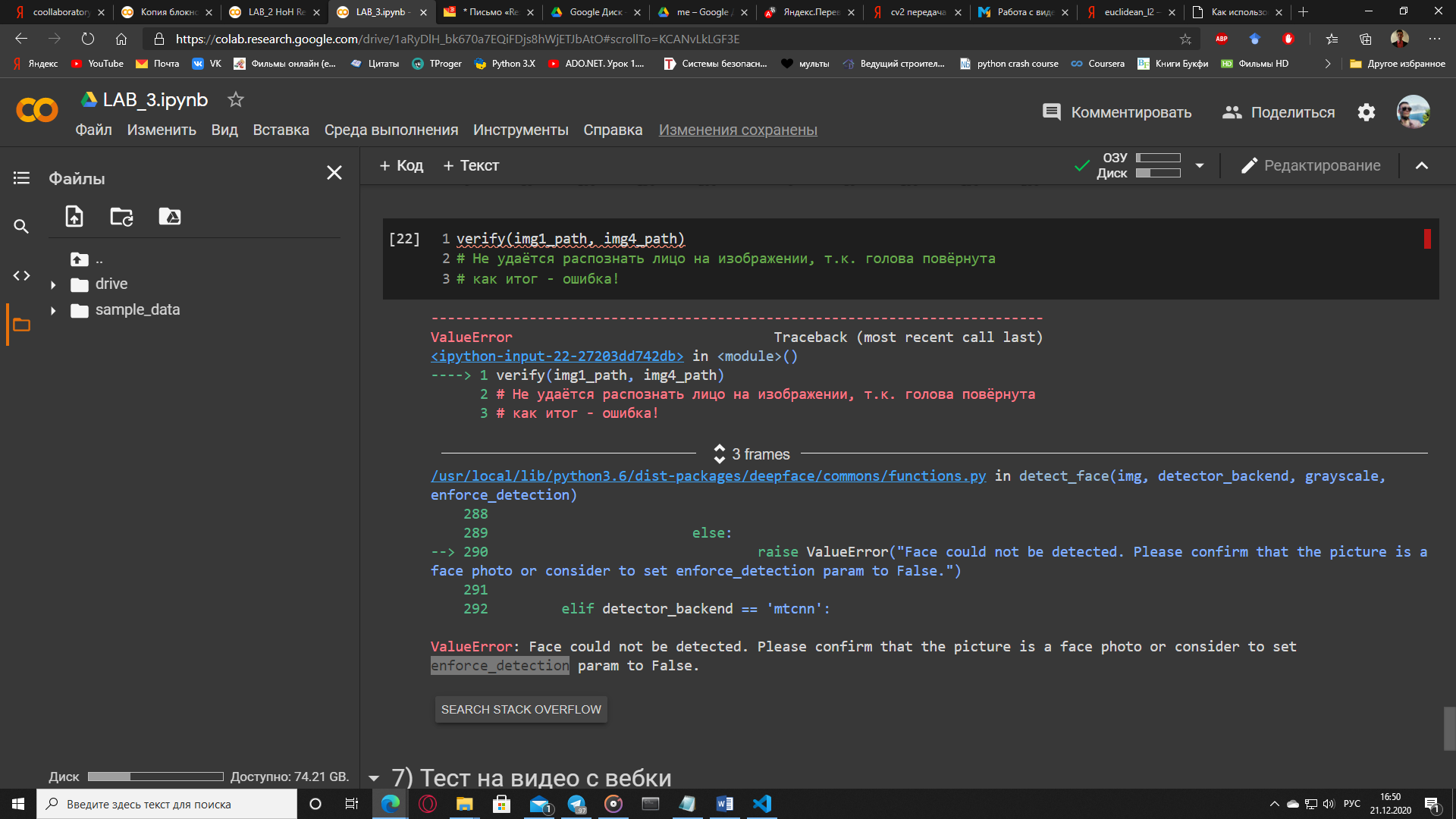


**Результат:** сеть выдала не совпадение. Евклидово расстояние 0.799. Порог по умолчанию стоит 0.55. Разница очень велика, потому что представлены 2 разных человека. Как видно евклидово расстояние чуть меньше (похоже в молодости я был чуть более на него похож :)), но сопоставимо с 0.83 как в прошлом эксперименте. Сеть справилась успешно.

**4 эксперимент:** я взял свою фотографию в анфас и ещё одну в профиль. На втором фото плохо видно лицо.



Сеть не смогла найти лицо на 2 изображении- выкинула ошибку.



Источники:

[deepface · PyPI](https://pypi.org/project/deepface/)