

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Факультет «Информационные технологии»

Кафедра «Искусственный интеллект и машинное обучение»

Лабораторная работа №16

Настройка работы с зависимостями

Автор:

Голиков Михаил Вячеславович, БВТ2402

Цель лабораторной работы

Настроить работу зависимостей, прописанных другими людьми из открытых источников, а также использовать библиотеку `face_recognition` на `python` для определения сходств лиц людей.

Ход выполнения лабораторной работы

Москва, 2025

Первым делом прочитаем инструкцию по установке библиотеки `face_recognition` на официальном сайте Python.

Installation

Requirements

- Python 3.3+ or Python 2.7
- macOS or Linux (Windows not officially supported, but might work)

Installing on Mac or Linux

First, make sure you have `dlib` already installed with Python bindings:

- [How to install dlib from source on macOS or Ubuntu](#)

Then, install this module from pypi using `pip3` (or `pip2` for Python 2):

```
pip3 install face_recognition
```

If you are having trouble with installation, you can also try out a [pre-configured VM](#).

Элемент 1 — Инструкция 1

Откроем переводчик и узнаем, что Windows – ПО не для программистов. Снесём Windows с компьютера, после чего установим один из множества дистрибутивов Linux (или откроем WSL). Перейдём на инструкцию для установки `dlib`.

How to install dlib v19.9 or newer (w/ python bindings) from github on macOS and Ubuntu

Pre-reqs:

- Have Python 3 installed. On macOS, this could be installed from homebrew or even via standard Python 3.6 downloaded installer from <https://www.python.org/download>. On Linux, just use your package manager.
- On macOS:
 - Install XCode from the Mac App Store (or install the XCode command line utils).
 - Have [homebrew](#) installed
- On Linux:
 - For a full list of apt packages required, check out the [example Dockerfile](#) and copy what's installed there.
 - These instructions assume you are using Ubuntu 16.04 or newer. If you are using 14.04, you can try [these installation instructions instead](#) to work around the old CMake version.
- These instructions assume you don't have an nVidia GPU and don't have Cuda and cuDNN installed and don't want GPU acceleration (since none of the current Mac models support this).

Clone the code from github:

```
git clone https://github.com/davisking/dlib.git
```

Build the main dlib library (optional if you just want to use Python):

```
cd dlib
mkdir build; cd build; cmake ..; cmake --build .
```

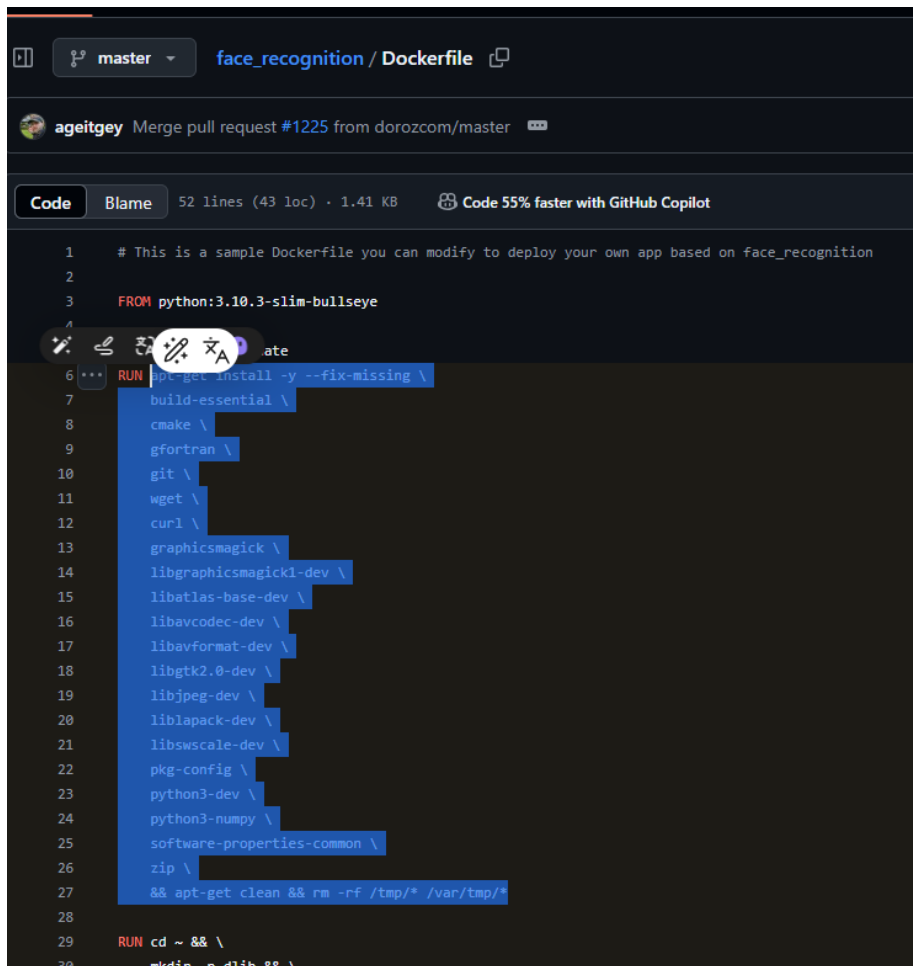
Build and install the Python extensions:

```
cd ..
python3 setup.py install
```

At this point, you should be able to run `python3` and type `import dlib` successfully.

Элемент 2 — Инструкция 2

Откроем третью инструкцию и скачаем вручную все нужные библиотеки. Далее в рекурсивном порядке выполним все инструкции всех библиотек, после чего скачаем саму face_recognition.



```
1 # This is a sample Dockerfile you can modify to deploy your own app based on face_recognition
2
3 FROM python:3.10.3-slim-bullseye
4
5
6 apt-get install -y --fix-missing \
7 build-essential \
8 cmake \
9 gfortran \
10 git \
11 wget \
12 curl \
13 graphicsmagick \
14 libgraphicsmagick1-dev \
15 libatlas-base-dev \
16 libavcodec-dev \
17 libavformat-dev \
18 libgtk2.0-dev \
19 libjpeg-dev \
20 liblapack-dev \
21 libswscale-dev \
22 pkg-config \
23 python3-dev \
24 python3-numpy \
25 software-properties-common \
26 zip \
27 && apt-get clean && rm -rf /tmp/* /var/tmp/*
28
29 cd ~ && \
30 mkdir -p .dlib && \
```

Элемент 3 — Инструкция 3

Далее напишем саму программу, которая будет проверять обе фотографии с оригиналом. Результат программы соответствует ожиданиям.

```
import face_recognition
known_face = face_recognition.load_image_file("base_image.jpg")
face_1 = face_recognition.load_image_file("face_1.jpg")
face_2 = face_recognition.load_image_file("face_2.jpg")

known_encoding = face_recognition.face_encodings(known_face)[0]
face_1_encoding = face_recognition.face_encodings(face_1)[0]
face_2_encoding = face_recognition.face_encodings(face_2)[0]

result_1 = face_recognition.compare_faces([known_encoding], face_1_encoding)
result_2 = face_recognition.compare_faces([known_encoding], face_2_encoding)

print(f"Первое лицо: {result_1}")
print(f"Второе лицо: {result_2}")
```

Элемент 4 — Код программы

```
root@DESKTOP-OUTQMH8:/mnt/e/VIIT_labs/Lab_16# python3 -m lab_16
Первое лицо: [True]
Второе лицо: [False]
root@DESKTOP-OUTQMH8:/mnt/e/VIIT_labs/Lab_16#
```

Элемент 5 — Результат программы



Элемент 6 — Фотографии для сравнения

Заключение

Был успешно изучен процесс работы с зависимостями. Все задачи выполнены, результаты соответствуют ожидаемым.