

This is for OpenPose 1.70 with Python.

System Configuration

Stuff you will need before downloading openpose,

OpenPose works best when running on GPU, so have all GPU installations. Don't use CPU mode as it is slow and comes with less functionality.

Downloads for GPU:

- CUDA version: 11.1.1
 - i. [CUDA Toolkit 11.1 Update 1 Downloads | NVIDIA Developer](#)
- cuDNN version: 8.0.5.39
 - i. [cuDNN Archive | NVIDIA Developer](#)

cuDNN will need a developer account to download:

Email: HATlabuvic@outlook.com

Password: standard HATlab password

It is good to know the GPU model incase errors occur.

GPU model: Nvidia GeForce RTX 3060 Laptop 6GB(if using the HATlaptop01 or HATlaptop02, if not check to find your GPU model)

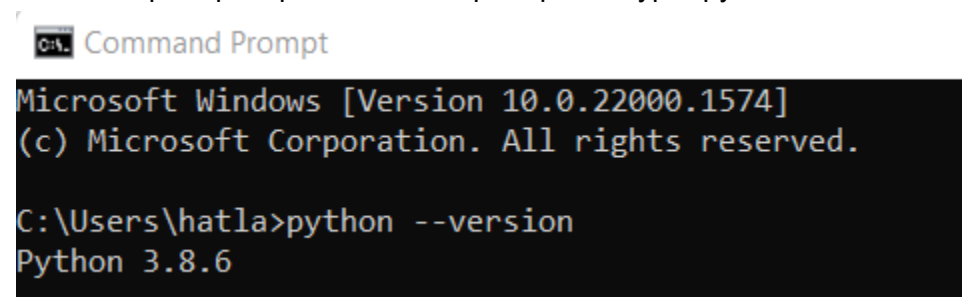
Download CMake to configure OpenPose build.(version 3.19.0 was recommended but newer versions also worked)

[Download | CMake](#)

Download Python version 3.8.6

[Python Release Python 3.8.6 | Python.org](#)

Once python is installed check to make sure it is the default python version running on command prompt. Open command prompt and type "python --version"

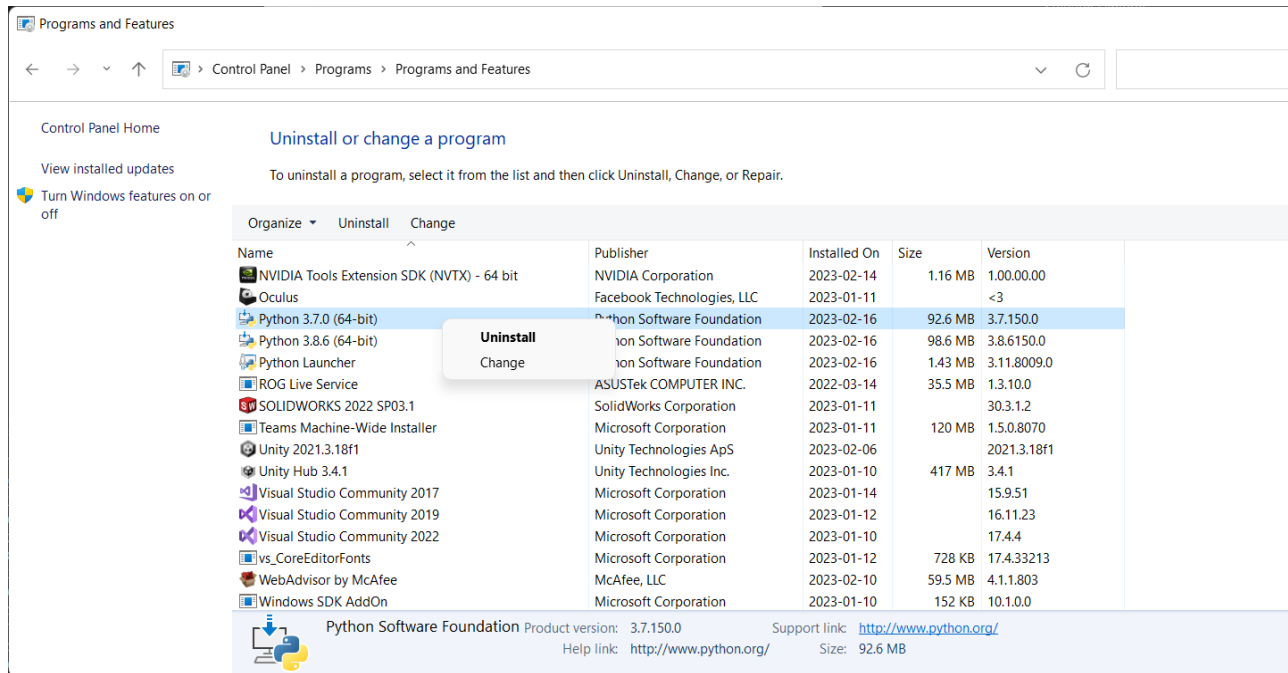


```
C:\> Command Prompt

Microsoft Windows [Version 10.0.22000.1574]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hatla>python --version
Python 3.8.6
```

If it returns Python 3.8.6, you are good to go! If not, go to the control panel and install all python versions.



Go to Uninstall a program(at the bottom-left of the control panel menu). From the list of programs click the python versions to get the option to uninstall them.

After they are all uninstalled, reinstall 3.8.6 and try again.

After you are on the correct version you will have to install some python packages.

1. Install numpy(numpy==1.19.3)

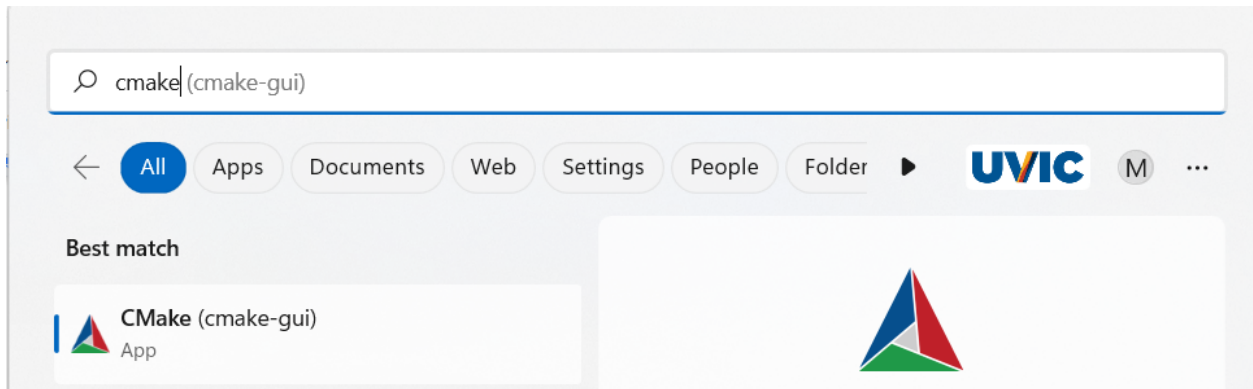
```
C:\Users\hatla>pip install numpy==1.19.3
```

2. Install cv2(called opencv-python when installing, but in files called cv2) (version: 4.4.0.40)

```
C:\Users\hatla>pip install opencv-python==4.4.0.40
```

Go to github and clone Openpose to your computer.

After python is fully configured, open cmake-gui.



Select the OpenPose root folder in the Where is the source code:

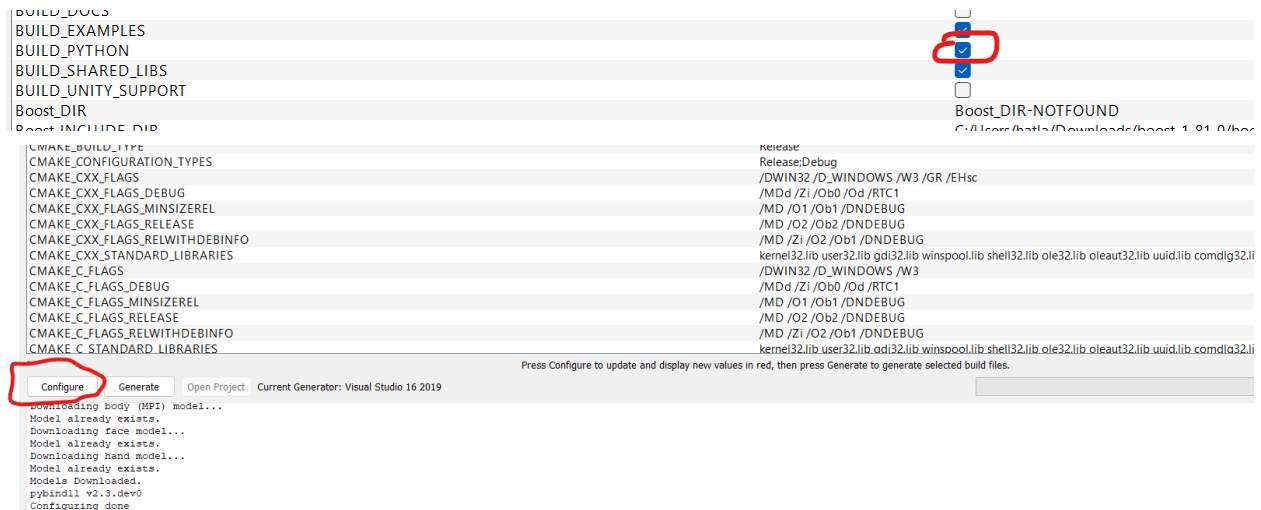
C:/Users/hatia/OneDrive/Documents/GitHub/openpose Browse Source...

Create a build folder in the OpenPose root folder and select it in the where to build binaries:

C:/Users/hatia/OneDrive/Documents/GitHub/openpose/build Browse Build...

If Prompted, select **Visual Studio Code 2019** and **x64** architecture

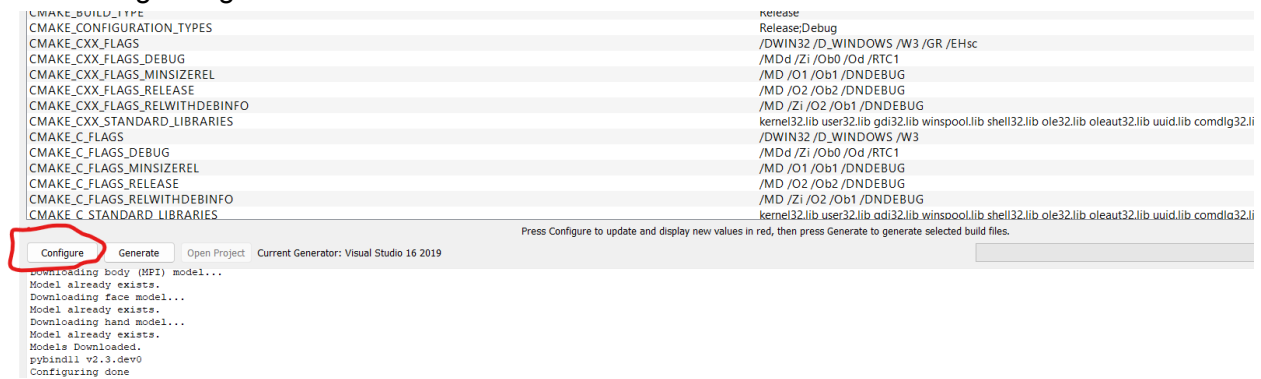
A list of options should appear, select **BUILD_PYTHON** and click configure, the configuration should take some time ~2mins



Some new options should appear, click **PYTHO_EXECUTABLE** and select the file path that matches the image below. Also make sure **PYBIND11_INSTALL** is checked.



Click configure again:

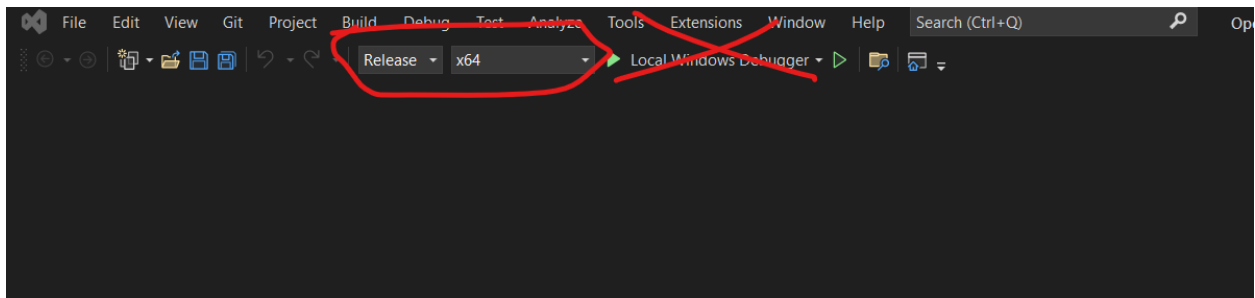


If options highlighted red appear, click configure until they stop appearing.

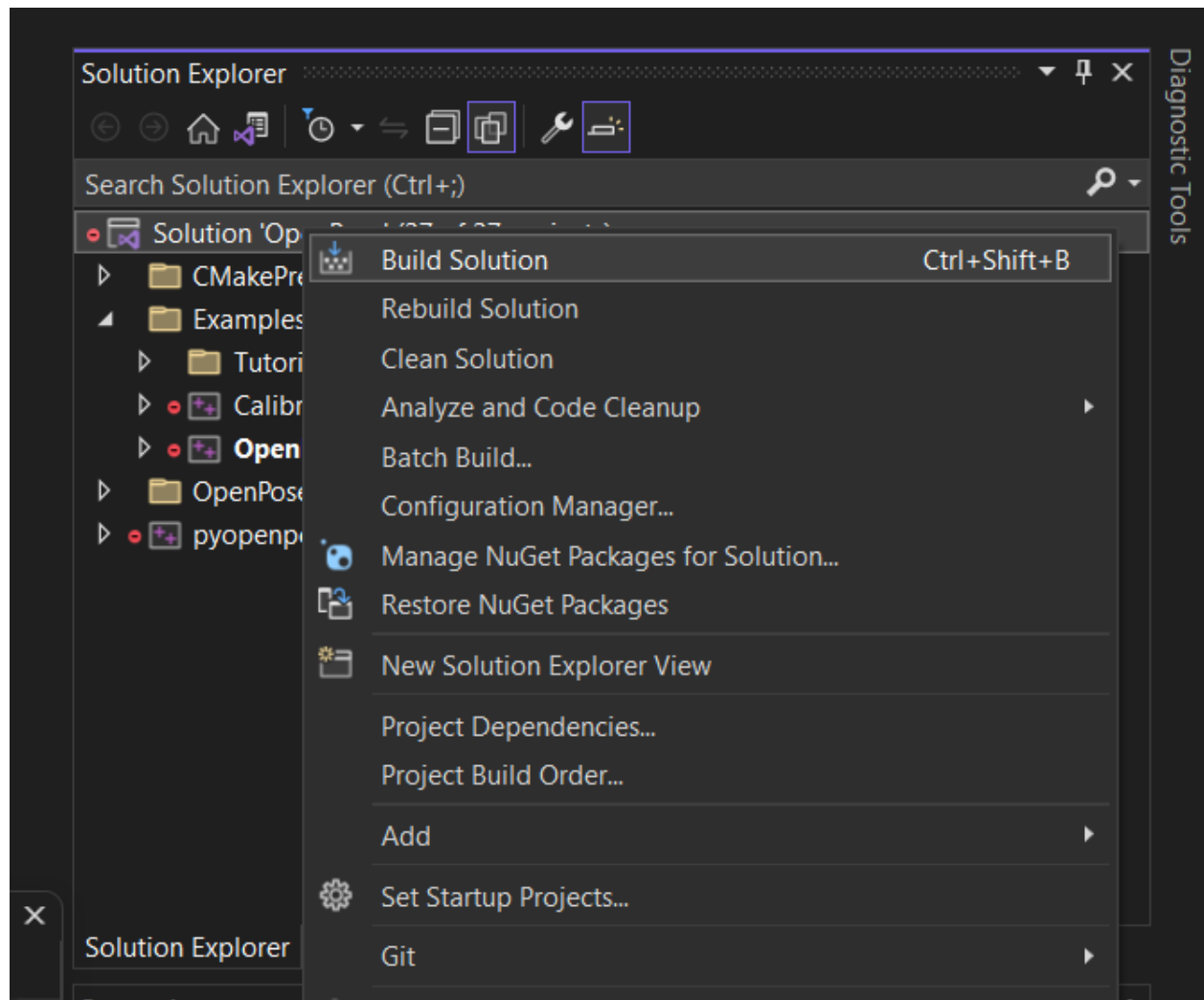
Next click **Generate**, the button beside Configure.

After the project is generated, click **Open Project**.

When visual studios has opened: select **Release** as the Solution Configuration and **x64** as the architecture, but **do not click the green play button!!**



Go to the solution explorer at the side of the page and right click on the solution tab, click “Build Solution”



After the Solution should build without any errors.

If you have an error message saying “**ssize_t is not identified**” or something similar, go to file explorer and open up the OpenPose root directory.

Go to “**openpose\3rdparty\pybind11\include\pybind11**” and open up **numpy.h**

```
C numpy.h 2 X
C: > Users > hatla > OneDrive > Documents > GitHub > openpose > 3rdparty > pybind11 > include > pybind11 > C numpy.h > ...
17 #include <cstdint>
18 #include <cstring>
19 #include <sstream>
20 #include <string>
21 #include <functional>
22 #include <utility>
23 #include <vector>
24 #include <typeindex>
25
26 #if defined(_MSC_VER)
27 # pragma warning(push)
28 # pragma warning(disable: 4127) // warning C4127: Conditional expression is constant
29 #endif
30
31 /* This will be true on all flat address space platforms and allows us to reduce the
32 whole npy_intp / ssize_t / Py_intptr_t business down to just ssize_t for all size
33 and dimension types (e.g. shape, strides, indexing), instead of inflicting this
34 upon the library user. */
35 static_assert(sizeof(size_t) == sizeof(Py_intptr_t), "size_t != Py_intptr_t");
```

change **ssize_t** to **size_t** on all instances of **ssize_t** in the document.

Now rebuild the solution, and make sure most of the processes succeed.

```
Output
Show output from: Build
5>Done building project "pyopenpose.vcxproj".
25> Creating library C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/examples/openpose/Release/OpenPoseDemo.lib and object C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/examples/openpose/Release/OpenPoseDemo.exe
25>OpenPoseDemo.vcxproj -> C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/x64/Release/OpenPoseDemo.exe
25>Done building project "OpenPoseDemo.vcxproj".
23> Creating library C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/examples/tutorial_api_cpp/Release/02_whole_body_from_image_default.lib and object C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/examples/tutorial_api_cpp/Release/02_whole_body_from_image_default.exe
23>02_whole_body_from_image_default.vcxproj -> C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/x64/Release/02_whole_body_from_image_default.exe
7> Creating library C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/examples/tutorial_api_cpp/Release/18_synchronous_custom_all_and_datum.lib and object C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/examples/tutorial_api_cpp/Release/18_synchronous_custom_all_and_datum.exe
7>18_synchronous_custom_all_and_datum.vcxproj -> C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/x64/Release/18_synchronous_custom_all_and_datum.exe
12> Creating library C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/examples/tutorial_api_cpp/Release/13_asynchronous_custom_input_output_and_datum.lib and object C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/examples/tutorial_api_cpp/Release/13_asynchronous_custom_input_output_and_datum.exe
12>13_asynchronous_custom_input_output_and_datum.vcxproj -> C:/Users/hatla/OneDrive/Documents/GitHub/openpose/build/x64/Release/13_asynchronous_custom_input_output_and_datum.exe
26>----- Rebuild All started: Project: ALL_BUILD, Configuration: Release x64 -----
27>----- Skipped Rebuild All: Project: INSTALL, Configuration: Release x64 -----
27>Project not selected to build for this solution configuration
===== Rebuild All: 26 succeeded, 0 failed, 1 skipped =====
===== Elapsed 01:24.062 =====
```

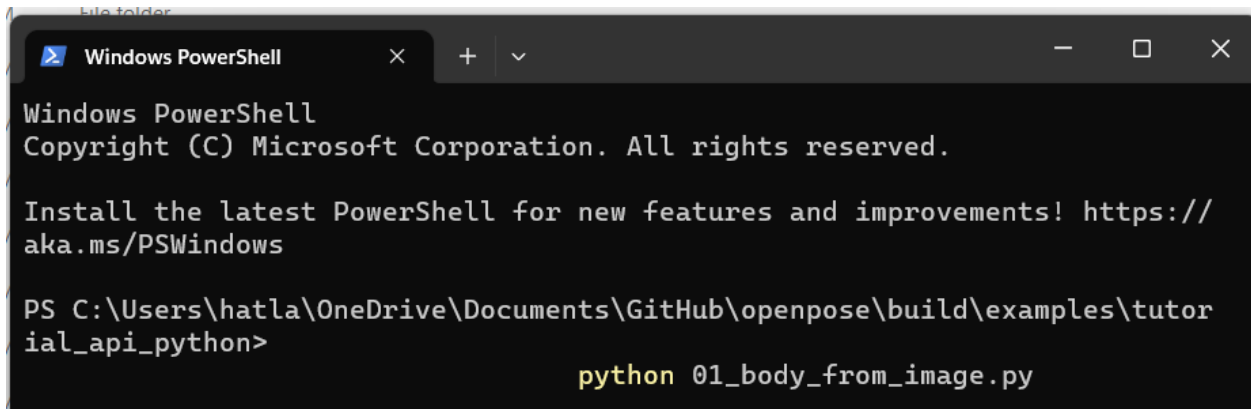
Go to the python api example folder and right click on the whitespace. Click open in Terminal.
Do not run these python files in visual studio or visual studio code as it won't work!!

Documents > GitHub > openpose > build > examples > tutorial_api_python

Name	Date modified	Type	Size
CMakeFiles	2023-02-16 1:54 PM	File folder	
01_body_from_image	2023-02-16 4:00 PM	Python Source File	3 KB
02_whole_body_from_image	2023-02-16 1:50 PM	Python Source File	4 KB
04_keypoints_from_images	2023-02-16 1:50 PM	Python Source File	4 KB
05_keypoints_from_images_multi_gpu	2023-02-16 1:50 PM	Python Source File	5 KB
06_face_from_image	2023-02-16 3:45 PM	Python Source File	4 KB
07_hand_from_image	2023-02-16 1:50 PM	Python Source File	4 KB
08_heatmaps_from_image	2023-02-16 1:50 PM	Python Source File	4 KB
09_keypoints_from_image		Python Source File	4 KB
cmake_install		CMake Source File	2 KB
INSTALL.vcxproj		VC++ Project	6 KB
INSTALL.vcxproj.filters		VC++ Project File...	1 KB
openpose_python_api		Python Source File	3 KB
pose_deploy.prototxt		PROTOTXT File	44 KB
pose_iter_58400.caffemodel		CAFFEMODEL File	102,262 KB

- View
- Sort by
- Group by
- Undo Rename Ctrl+Z
- New
- Properties Alt+Enter
- Open in Terminal
- Show more options Shift+F10

To test if the API is working correctly, type “**python 01_body_from_image.py**”

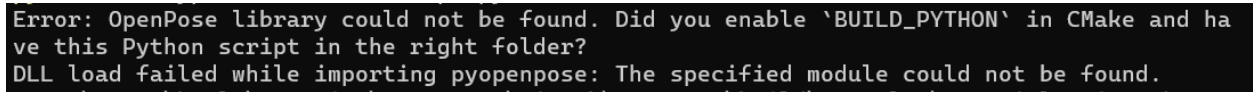


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

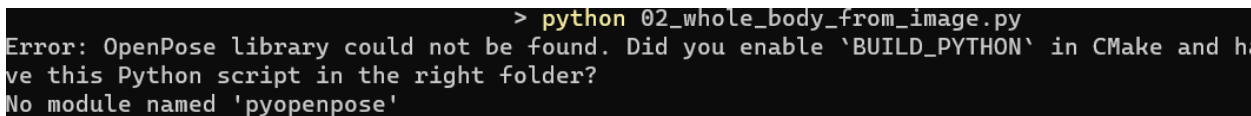
PS C:\Users\hatla\OneDrive\Documents\GitHub\openpose\build\examples\tutorial_api_python> python 01_body_from_image.py
```

If you get “OpenPose library could not be found” error



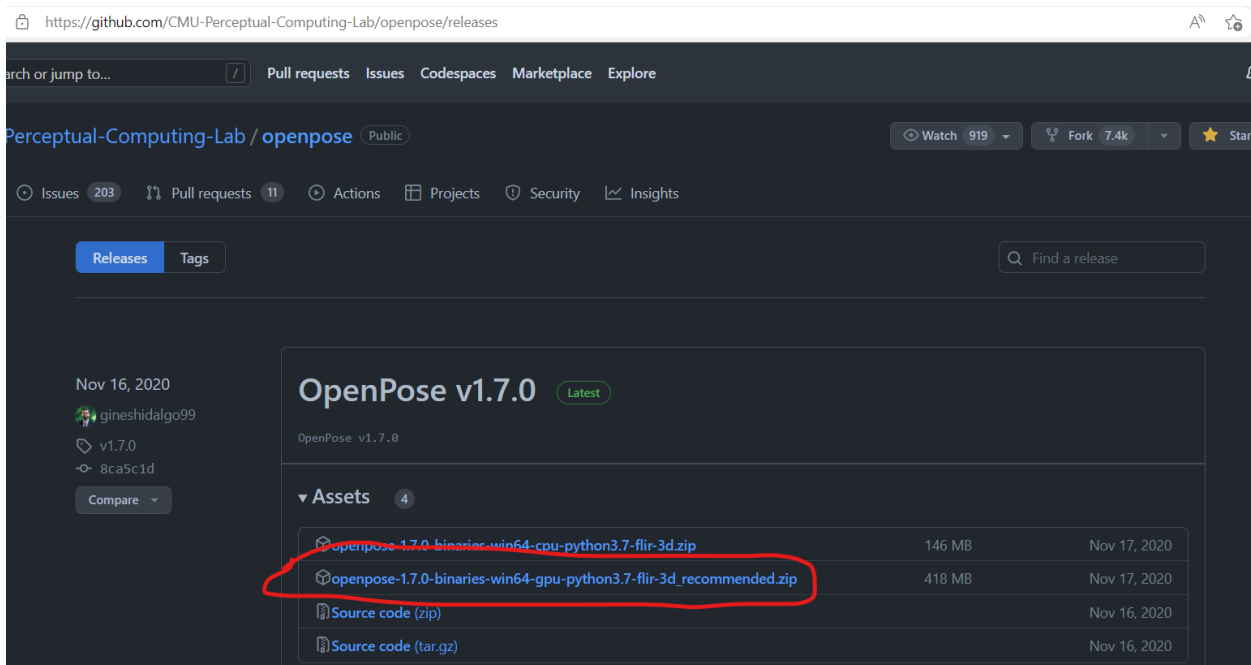
```
Error: OpenPose library could not be found. Did you enable 'BUILD_PYTHON' in CMake and have this Python script in the right folder?
DLL load failed while importing pyopenpose: The specified module could not be found.
```

Or if you had the “no module named ‘pyopenpose’” error



```
> python 02_whole_body_from_image.py
Error: OpenPose library could not be found. Did you enable 'BUILD_PYTHON' in CMake and have this Python script in the right folder?
No module named 'pyopenpose'
```

Go back to the OpenPose github page and go to the releases.



https://github.com/CMU-Perceptual-Computing-Lab/openpose/releases

Perceptual-Computing-Lab / openpose Public

Issues 203 Pull requests 11 Actions Projects Security Insights

Releases Tags Find a release

Nov 16, 2020 gineshidalgo99 v1.7.0 8ca5c1d Compare

OpenPose v1.7.0 Latest

OpenPose v1.7.0

Assets 4

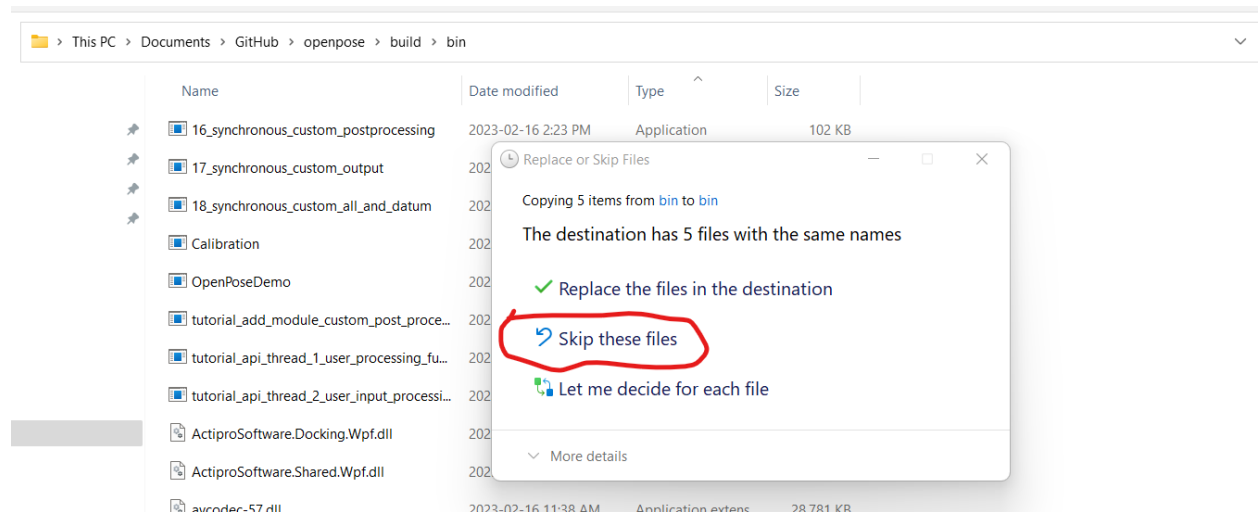
Asset	Size	Created
openpose-1.7.0-binaries-win64-cpu-python3.7-flir-3d.zip	146 MB	Nov 17, 2020
openpose-1.7.0-binaries-win64-gpu-python3.7-flir-3d_recommended.zip	418 MB	Nov 17, 2020
Source code (zip)		Nov 16, 2020
Source code (tar.gz)		Nov 16, 2020

Download “openpose-1.7.0-binaries-win64-gpu-python3.7-flir-3d_recommended.zip”. Once downloaded, extract all to a folder in Downloads.

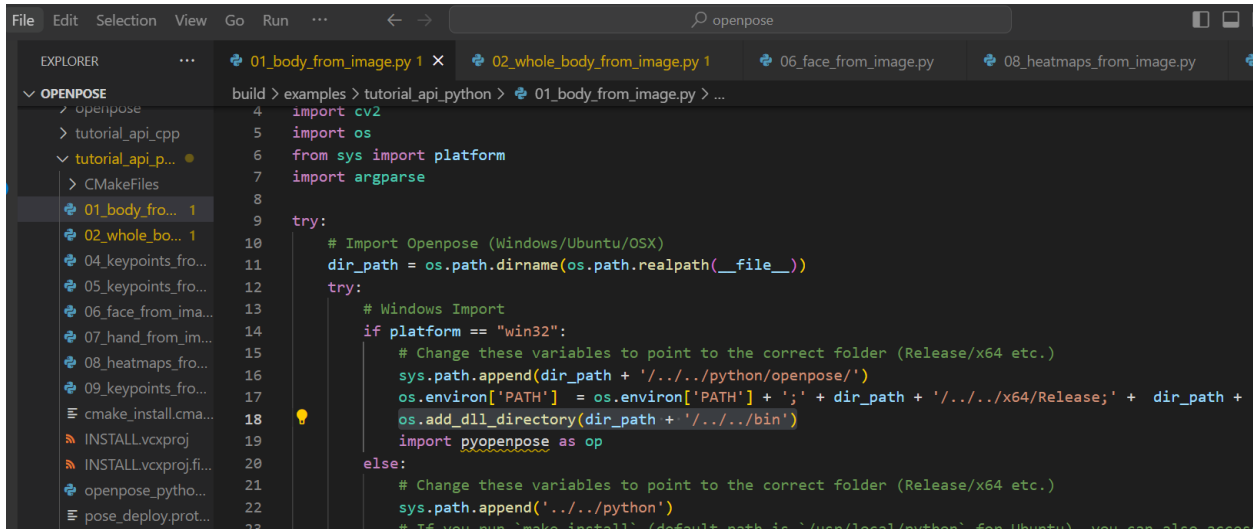
Downloads > openpose-1.7.0-binaries-win64-gpu-python3.7-flir-3d_recommended > openpose > bin

Name	Date modified	Type	Size
14_synchronous_custom_input	2023-02-16 11:38 AM	Application	95 KB
15_synchronous_custom_preprocessing	2023-02-16 11:38 AM	Application	96 KB
16_synchronous_custom_postprocessing	2023-02-16 11:38 AM	Application	96 KB
17_synchronous_custom_output	2023-02-16 11:38 AM	Application	107 KB
18_synchronous_custom_all_and_datum	2023-02-16 11:38 AM	Application	581 KB
ActiproSoftware.Docking.Wpf.dll	2023-02-16 11:38 AM	Application extens...	644 KB
ActiproSoftware.Shared.Wpf.dll	2023-02-16 11:38 AM	Application extens...	2,001 KB
avcodec-57.dll	2023-02-16 11:38 AM	Application extens...	28,781 KB
avdevice-57.dll	2023-02-16 11:38 AM	Application extens...	1,402 KB
avfilter-6.dll	2023-02-16 11:38 AM	Application extens...	4,173 KB
avformat-57.dll	2023-02-16 11:38 AM	Application extens...	6,753 KB
avutil-55.dll	2023-02-16 11:38 AM	Application extens...	584 KB
boost_date_time-vc142-mt-gd-x64-1_74...	2023-02-16 11:38 AM	Application extens...	15 KB
boost_date_time-vc142-mt-x64-1_74.dll	2023-02-16 11:38 AM	Application extens...	10 KB
boost_filesystem-vc142-mt-gd-x64-1_74...	2023-02-16 11:38 AM	Application extens...	208 KB
boost_filesystem-vc142-mt-x64-1_74.dll	2023-02-16 11:38 AM	Application extens...	115 KB
boost_system-vc142-mt-gd-x64-1_74.dll	2023-02-16 11:38 AM	Application extens...	15 KB
boost_system-vc142-mt-x64-1_74.dll	2023-02-16 11:38 AM	Application extens...	10 KB
boost_thread-vc142-mt-gd-x64-1_74.dll	2023-02-16 11:38 AM	Application extens...	174 KB

Go to the bin folder and copy all .dll files and move them to your main OpenPose bin folder, Skip all files that have the same names.

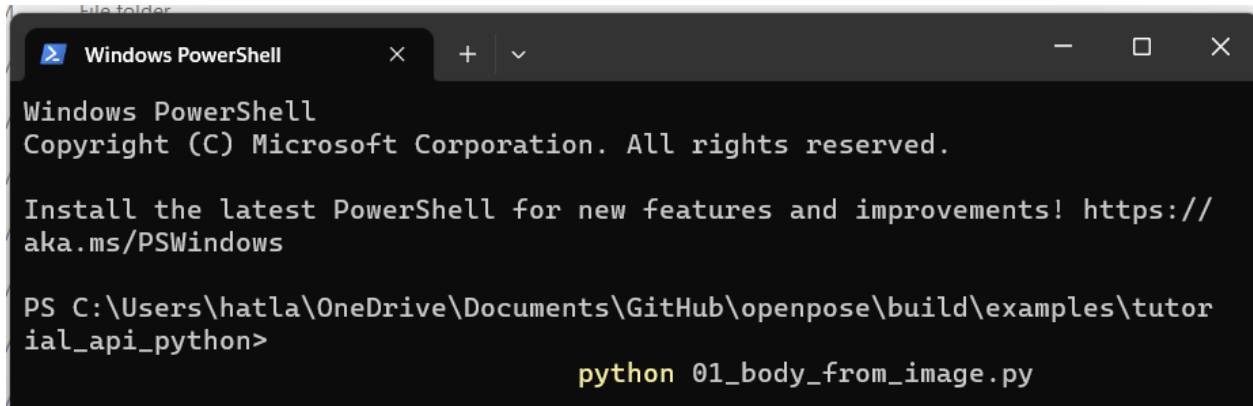


For each test file you will have to add “`os.add_dll_directory(dir_path + '/../bin')`” before importing `pyopenpose`.



```
File Edit Selection View Go Run ... openpose
EXPLORER
  OPENPOSE
    openpose
    tutorial_api_cpp
    tutorial_api_p...
    CMakeFiles
    01_body_fro... 1
    02_whole_bo... 1
    04_keypoints_fro...
    05_keypoints_fro...
    06_face_from_ima...
    07_hand_from_im...
    08_heatmaps_fro...
    09_keypoints_fro...
    cmake_install.cma...
    INSTALL.vcxproj
    INSTALL.vcxproj.f...
    openpose_pytho...
    pose_deploy.prot...
  build > examples > tutorial_api_python > 01_body_from_image.py > ...
4 import cv2
5 import os
6 from sys import platform
7 import argparse
8
9 try:
10     # Import Openpose (Windows/Ubuntu/OSX)
11     dir_path = os.path.dirname(os.path.realpath(__file__))
12     try:
13         # Windows Import
14         if platform == "win32":
15             # Change these variables to point to the correct folder (Release/x64 etc.)
16             sys.path.append(dir_path + '/../python/openpose/')
17             os.environ['PATH'] = os.environ['PATH'] + ';' + dir_path + '/../x64/Release;' + dir_path +
18             os.add_dll_directory(dir_path + '/../bin')
19             import pyopenpose as op
20         else:
21             # Change these variables to point to the correct folder (Release/x64 etc.)
22             sys.path.append('../python')
23             # If you run `make install` (default path is `/usr/local/python` for Ubuntu) you can also access
```

Try running “`python 01_body_from_image.py`” again



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hatla\OneDrive\Documents\GitHub\openpose\build\examples\tutorial_api_python> python 01_body_from_image.py
```

If the picture below opens congratulation you configured it correctly!

