

1. Run the following from <https://brew.sh/>:

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

2. Install a Mac OS X dependencies

```
brew install coreutils python3 cmake libusb wget opencv
```

3. Now that python3 is installed via brew, set the path to access it so that the system python isn't used by pip3 and so on.

Run the following (from [here](#), thanks Satya!)

```
echo "# Homebrew" >> ~/.bash_profile  
echo "export PATH=/usr/local/bin:$PATH" >> ~/.bash_profile  
source ~/.bash_profile
```

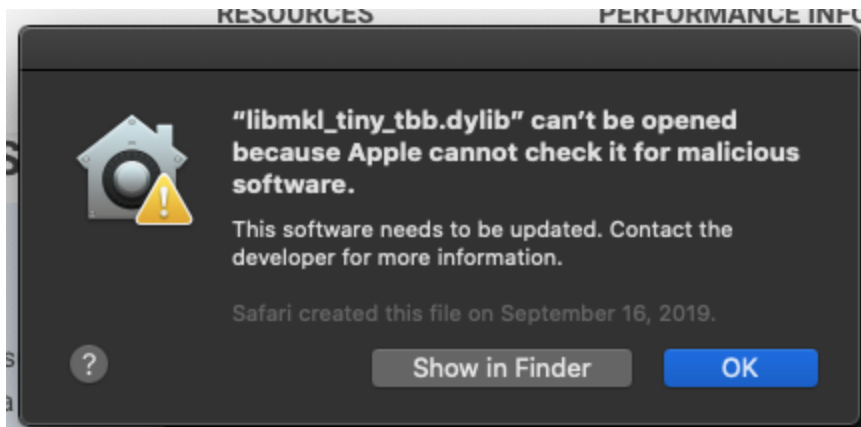
4. Install python-opencv

```
pip3 install opencv-python
```

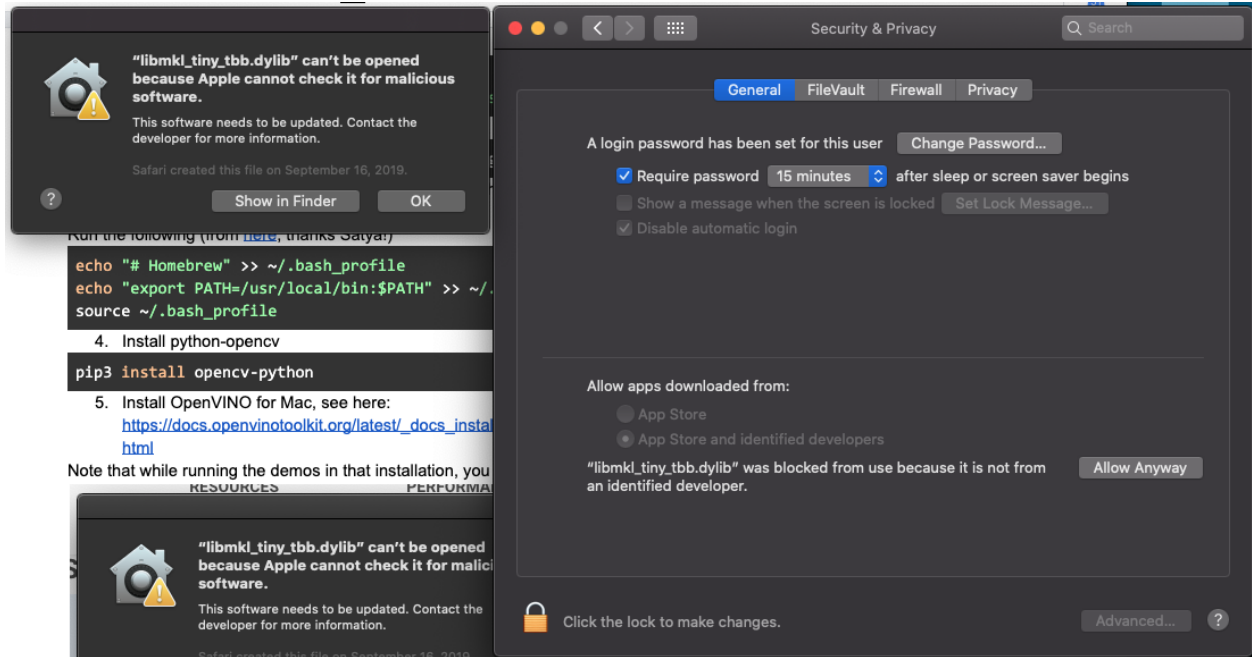
5. Install OpenVINO for Mac, see here:

https://docs.openvino toolkit.org/latest/_docs_install_guides_installing_openvino_macos.html

Note that while running the demos in that installation, you will likely see this:

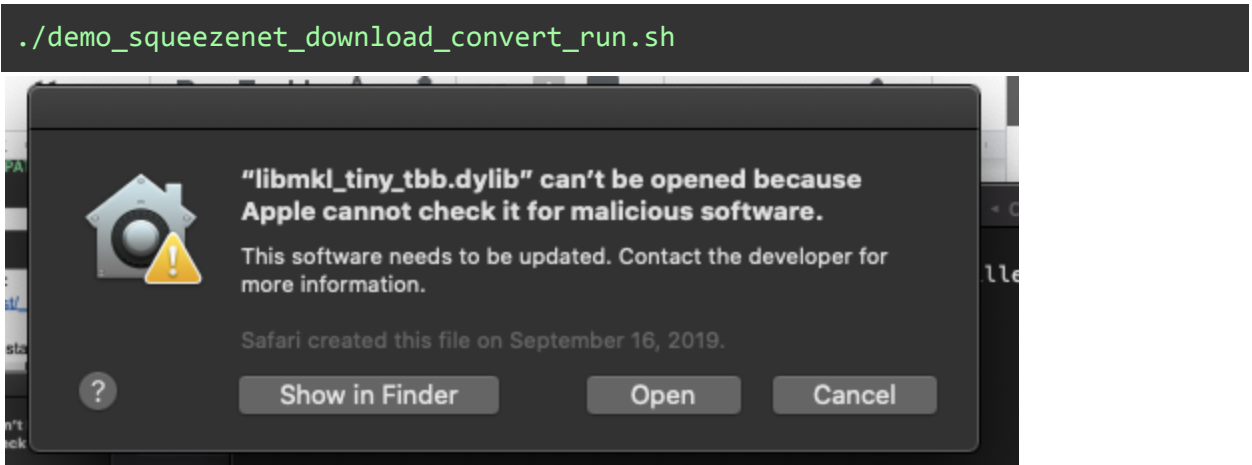


Go to Security & Privacy under System Preferences and click 'Allow Anyway' and then click OK.



You likely will need to click "Allow Anyway" multiple times.

This process will cause the demo to fail. Run it again after having gone through the above process and now you'll be prompted to allow this dylib to run, click open on this window.



This demo and the subsequent OpenVINO demos should now run:

```
demo — -bash — 121x24

classid probability label
-----
817 0.6853041 sports car, sport car
479 0.1835186 car wheel
511 0.0917199 convertible
436 0.0200693 beach wagon, station wagon, wagon, estate car, beach waggon, station waggon, waggon
751 0.0069604 racer, race car, racing car
656 0.0044177 minivan
717 0.0024739 pickup, pickup truck
581 0.0017788 grille, radiator grille
468 0.0013083 cab, hack, taxi, taxicab
661 0.0007443 Model T

[ INFO ] Execution successful

[ INFO ] This sample is an API example, for any performance measurements please use the dedicated benchmark_app tool

#####

Demo completed successfully.
```

6. Clone the DepthAI repository && change directory into this repository.

```
cd ~
git clone https://github.com/luxonis/depthai-api.git && cd depthai-api
```

7. Install depthai dependencies.

```
./install_dependencies.sh
```

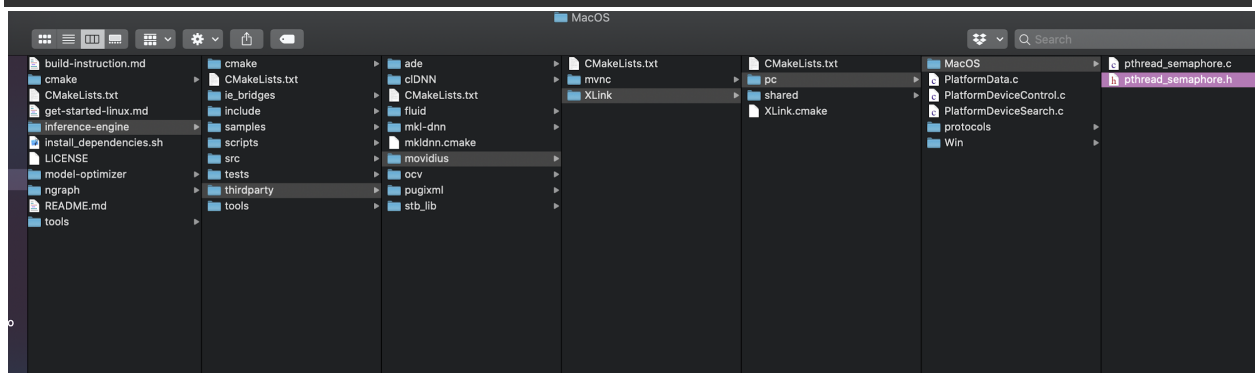
8. Checkout OpenVINO 2020.1 from the shared libraries (note that future DepthAI which natively uses 2020.1 will not need this).

```
cd shared/3rdparty/dldt && git checkout 2020.1
```

9. Now we're ready to build DepthAI for Mac, but it seems like the pthread_semaphore.h is an unexpected place on the Mac. So copying over this file to the location that the build scripts expect is a work around for now until we update the build script.

Below is where the pthread_semaphore.h actually exists on the Mac:

```
~/depthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/pc/MacOS
```



And from the error above, it's wanting to find this .h file here:

```
~/depthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/shared/include/
```

So let's copy it there as a workaround (only the .h otherwise the .c gets built twice:

```
cd
~/depthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/pc/MacOS
cp pthread_semaphore.h
~/depthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/shared/include/
```

10. Now that this is copied over, run `./build_py_module.sh`, it will now fail on `ld: library not found for -lusb-1.0`

```
cd ~/depthai-api && ./build_py_module.sh
```

To solve this problem, I managed to work around it by modifying the `link.txt` file located at `~/depthai-api/host/py_module/build/CMakeFiles/depthai.dir` and adding the additional flag `-L /usr/local/lib`, after which compilation succeeded.

```
cd ~/depthai-api/host/py_module/build/CMakeFiles/depthai.dir
vim link.txt
end
i
-L /usr/local/lib
esc:wq
```

Then to check it:

```
[DESKTOP-0GSI4CI:depthai.dir leeroy$ cat link.txt
/Library/Developer/CommandLineTools/usr/bin/c++ -isysroot /Library/Developer/CommandLineTools/SDKs
/MacOSX10.15.sdk -bundle -Wl,-headerpad_max_install_names -o depthai.cpython-37m-darwin.so CMakeFi
les/depthai.dir/py_bindings.cpp.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/host/core/device_
support_listener.cpp.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/host/core/pipeline/host_pipe
line.cpp.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/host/core/pipeline/host_pipeline_config.
cpp.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/host/core/pipeline/cnn_host_pipeline.cpp.o CM
akeFiles/depthai.dir/Users/leeroy/depthai-api/host/core/types.cpp.o CMakeFiles/depthai.dir/Users/le
eroy/depthai-api/host/core/host_json_helper.cpp.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/h
ost/core/host_data_reader.cpp.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/host/core/disparity
_stream_post_processor.cpp.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/json_helper.cpp
.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/stream/stream_info.cpp.o CMakeFiles/depth
ai.dir/Users/leeroy/depthai-api/shared/xlink/xlink_wrapper.cpp.o CMakeFiles/depthai.dir/Users/leero
y/depthai-api/shared/disparity_luts.cpp.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/3r
dparty/dldt/inference-engine/thirdparty/movidius/XLink/pc/PlatformData.c.o CMakeFiles/depthai.dir/U
sers/leeroy/depthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/pc/Platform
DeviceControl.c.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/3rdparty/dldt/inference-en
gine/thirdparty/movidius/XLink/pc/PlatformDeviceSearch.c.o CMakeFiles/depthai.dir/Users/leeroy/dept
hai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/pc/protocols/pcie_host.c.o
CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/mo
vidius/XLink/pc/protocols/usb_boot.c.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/3rdpa
rty/dldt/inference-engine/thirdparty/movidius/XLink/shared/src/XLinkData.c.o CMakeFiles/depthai.dir
/Users/leeroy/depthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/shared/sr
c/XLinkDeprecated.c.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/3rdparty/dldt/inferenc
e-engine/thirdparty/movidius/XLink/shared/src/XLinkDevice.c.o CMakeFiles/depthai.dir/Users/leeroy/d
epthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/shared/src/XLinkDispatch
er.c.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/3rdparty/dldt/inference-engine/thirdp
arty/movidius/XLink/shared/src/XLinkDispatcherImpl.c.o CMakeFiles/depthai.dir/Users/leeroy/depthai-
api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/shared/src/XLinkPrivateDefines.
c.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/3rdparty/dldt/inference-engine/thirdpart
y/movidius/XLink/shared/src/XLinkPrivateFields.c.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/
shared/3rdparty/dldt/inference-engine/thirdparty/movidius/XLink/shared/src/XLinkStream.c.o CMakeFil
es/depthai.dir/Users/leeroy/depthai-api/shared/3rdparty/dldt/inference-engine/thirdparty/movidius/X
Link/shared/src/XLinkStringUtils.c.o CMakeFiles/depthai.dir/Users/leeroy/depthai-api/shared/3rdpart
y/dldt/inference-engine/thirdparty/movidius/XLink/pc/MacOS/pthread_semaphore.c.o -undefined dynamic
_lookup -flto -lusb-1.0 -lpthread json-schema-validator/libnlohmann_json_schema_validator.a -L /usr
/local/lib
```

11. Now that this is there, try building DepthAI again:

```
cd ~/depthai-api && ./build_py_module.sh
```

This produces a bazillion ld: warnings (and I don't know what these are about):

[illegible]

But it does build:

```
[100%] Built target depthai
[BUILD host/py_module/build] cp depthai.cpython-37m-darwin.so ../../../../../../
```

12. Now that this is built, let's copy the .so over to the depthai-python-extras. I had checked out depthai-api by itself, so I need to now clone depthai-python-extras and then copy over the .so.

```
cd ~
git clone https://github.com/luxonis/depthai-python-extras.git
cd depthai-python-extras
cp ../depthai-api/host/py_module/build/depthai.cpython-37m-darwin.so .
```

13. Now that this is copied over, you can plug in your DepthAI or microAI and run test.py from the depthai-python-extras:

```
python3 test.py
```