**TensorFlow Windows C++ Walk-through**

**Instructions for compiling TensorFlow on Windows 10 with C++ and running a first TensorFlow C++ program**

**Walk-through video:**

<https://www.youtube.com/watch?v=oXpsAiSajE0>

**Table of Contents:**

1) Complete the previous walk-throughs

2) Complete the previous walk-through

3) Clone the repository containing this document

4) Download images

4a) Download Google flower images

4b) Download your own images

5) Separate out some test images

6) Run ***retrain.py***

7) Run ***test.py***

----------------------------------------------------------------------------------------

**1) Complete the previous walk-throughs**

Complete this previous walk-through if you have not already:

<https://www.youtube.com/watch?v=oXWVuK_NjbY>

<https://github.com/MicrocontrollersAndMore/TensorFlow_Tut_1_Installation_and_First_Progs>

**ToDo: put links to 2 here**

**ToDo: put links to 3 here**

**2) Clone the repository containing this document**

Clone the repository containing this document:

<https://github.com/MicrocontrollersAndMore/TensorFlow_Tut_2_Classification_Walk-through>

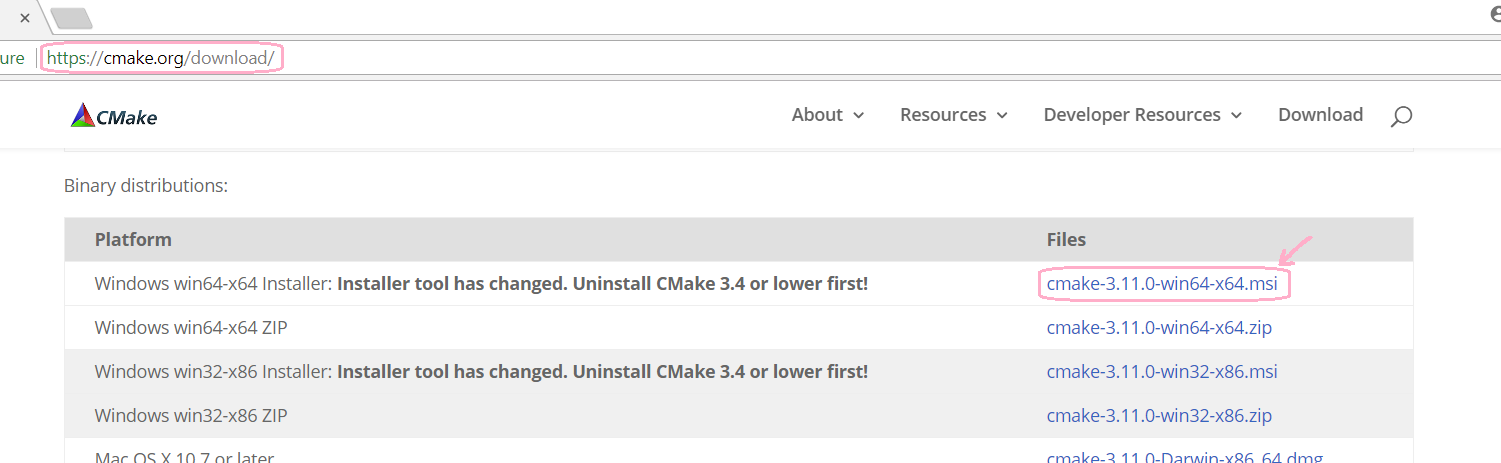
to a location you’d like to work in, for example I’ll use:

C:\Users\cdahms\Documents\TensorFlow\_Tut\_2\_Classification\_Walk-through

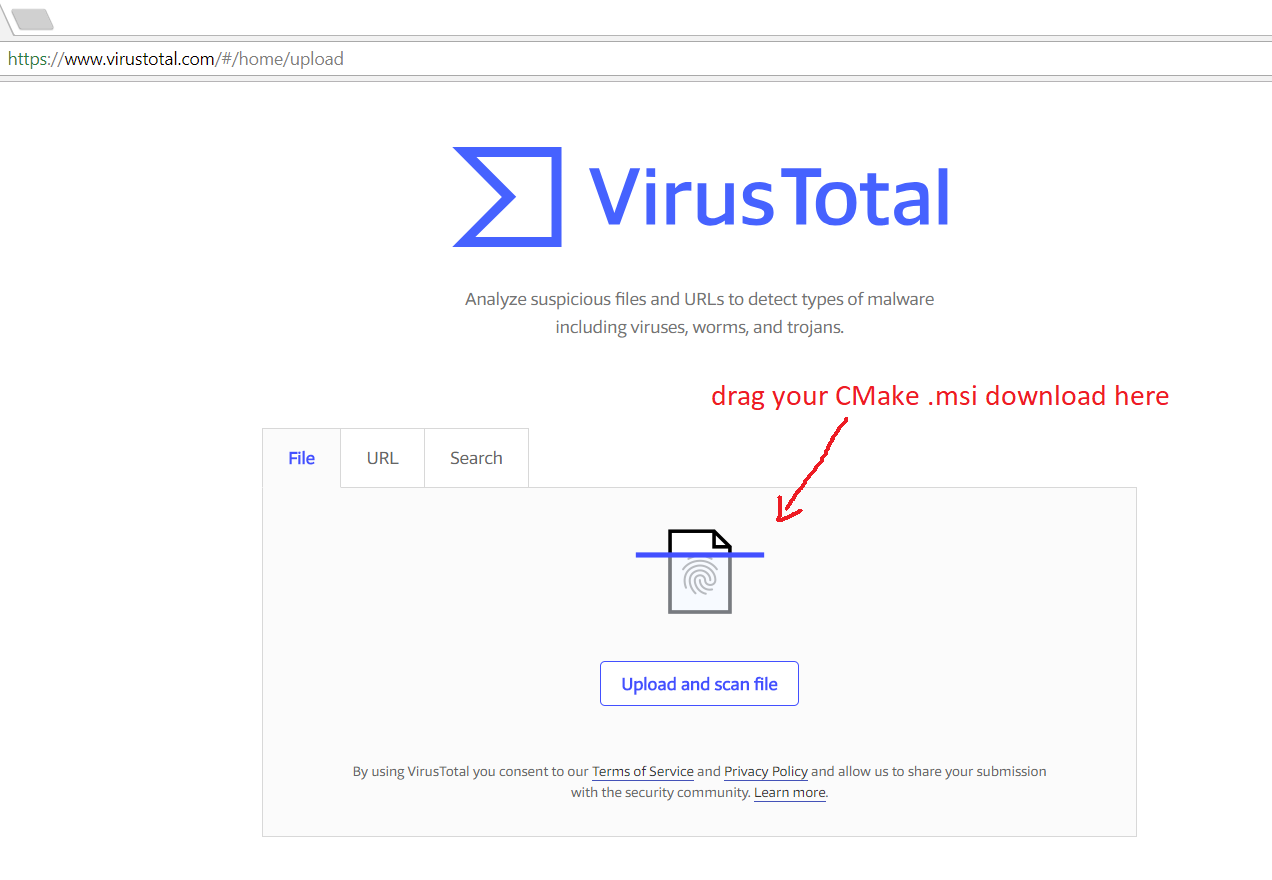
You can choose any directory you’d like. Going forward in this document we’ll refer to this location as ***(repository\_location)***

**3) Download and Install CMake**

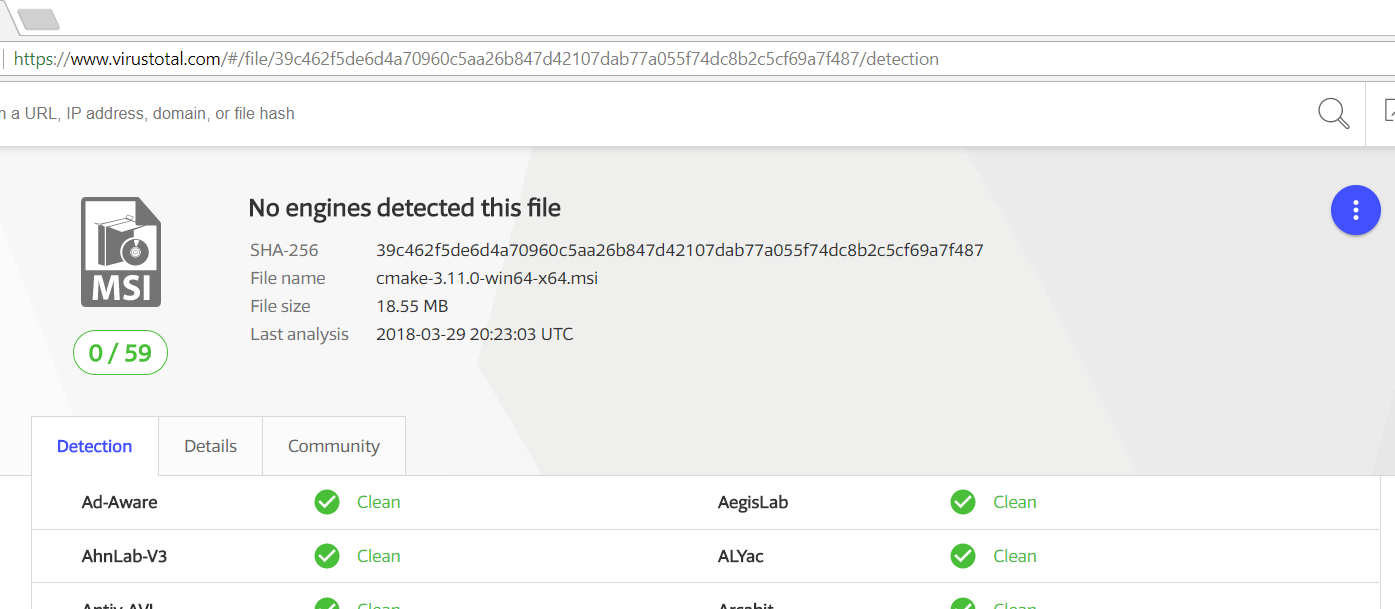
Go to the <https://cmake.org> and download the latest Windows x64 installer:

****

In my case Windows 10 suggested the CMake download may be a virus, don’t worry, CMake is not a virus. If you’re in doubt, go to <https://www.virustotal.com> and drag CMake to the “Upload and scan file” box:

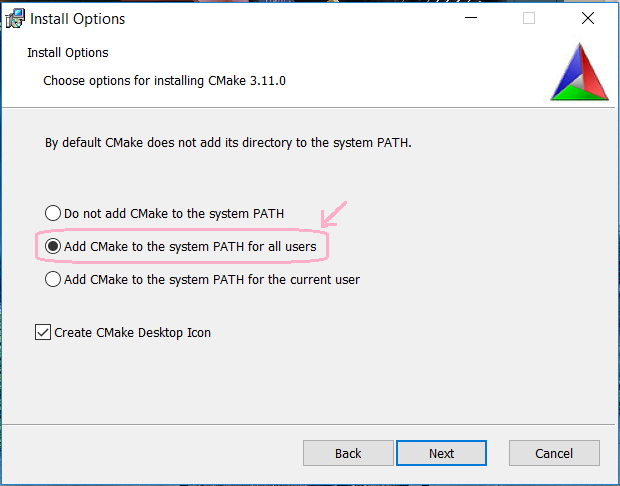


You will see CMake is not a virus:



Click through the virus warning Windows 10 may show you and proceed with the CMake x64 .msi install.

When asked, choose “Add CMake to the system PATH for all users”:



Otherwise you can choose all the defaults. Once the CMake install is complete, reboot (do NOT skip the reboot).

**4) Install Git**

If you’re following this walk-through you probably already have Git installed, but in case you don’t go ahead and install Git now:

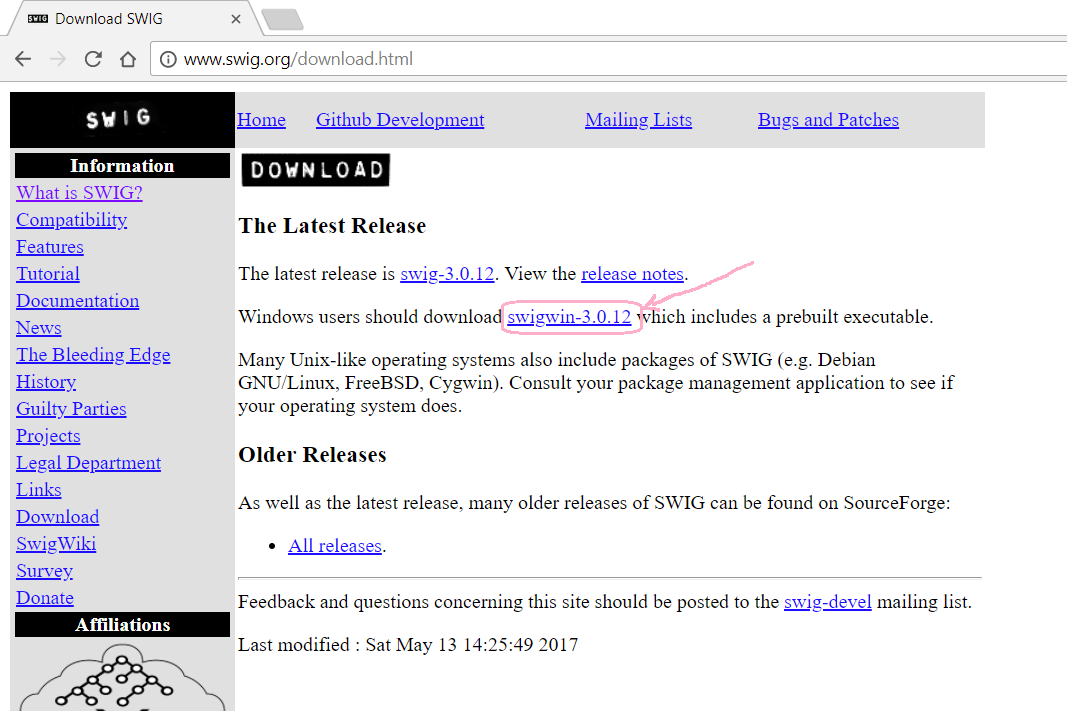


In most cases the default options should be ok. If asked, choose to add Git to the PATH for all users. Once the Git install is complete, reboot (do NOT skip the reboot).

**5) Install SWIG**

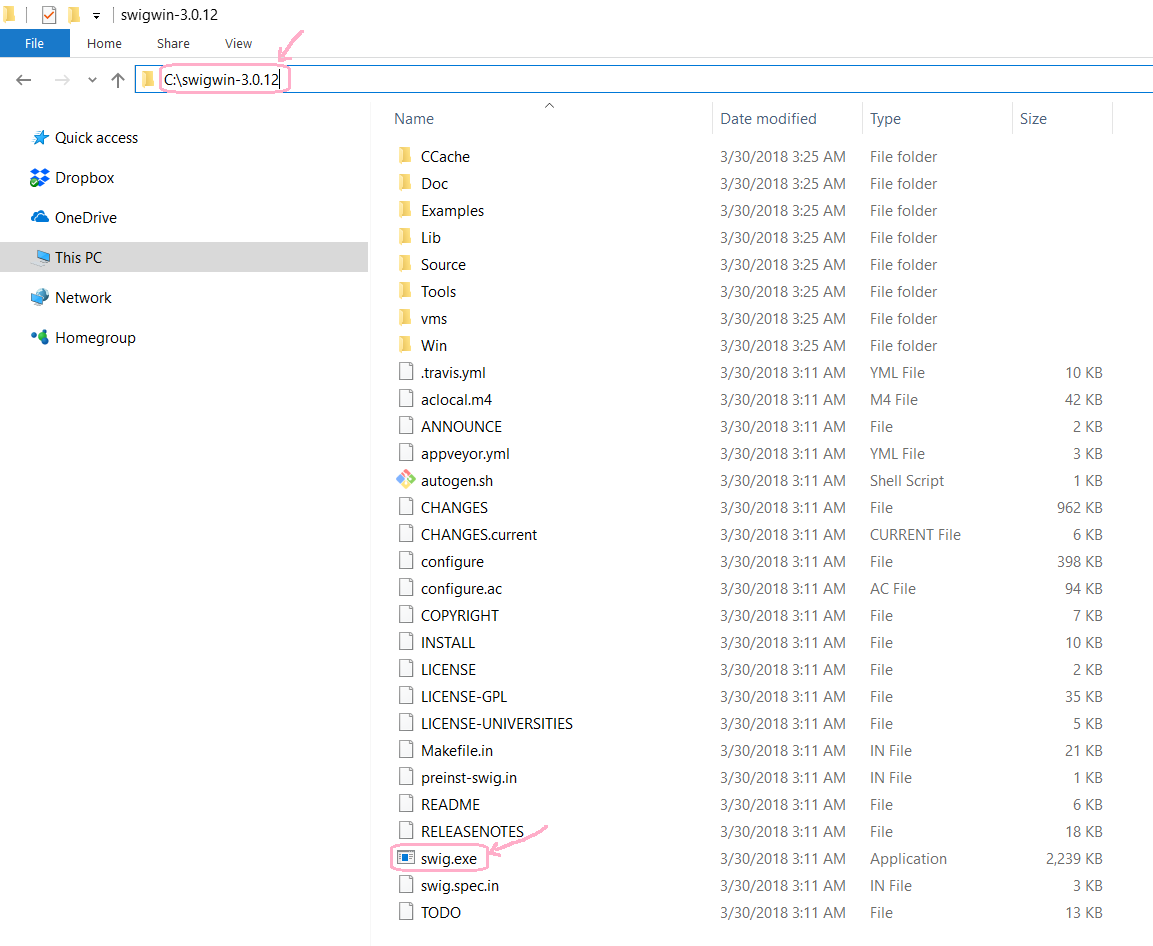
SWIG is short for Simplified Wrapper and Interface Generator, you can read more about SWIG on the SWIG Wikipedia page <https://en.wikipedia.org/wiki/SWIG>

Download the Windows version of SWIG, “SWIGWin”:



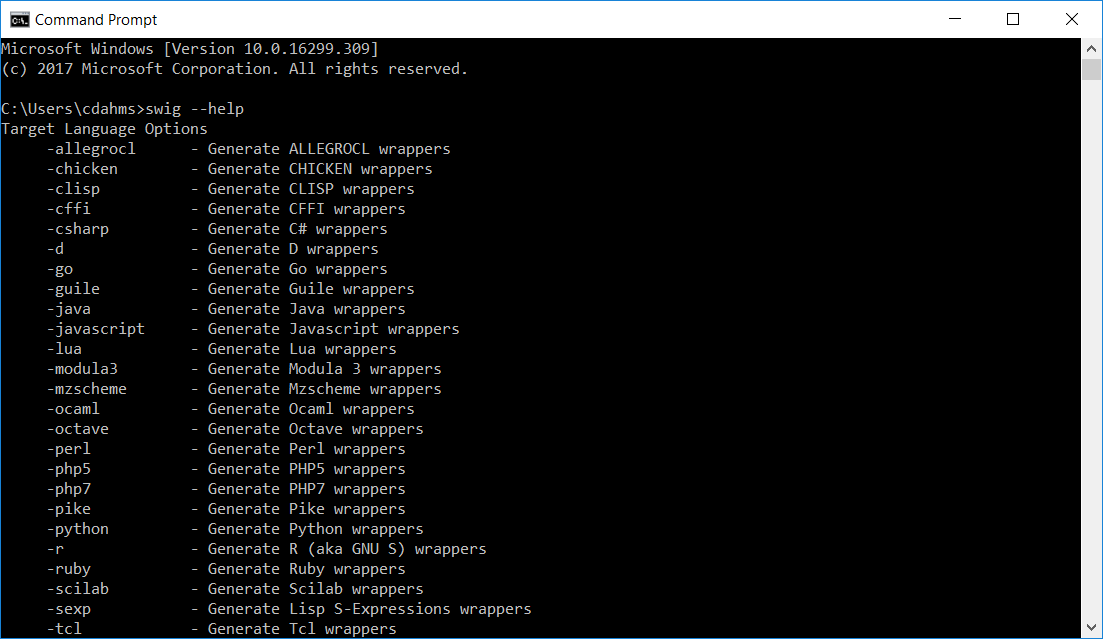
There isn’t an install for SWIG Win, so simply extract the .zip and copy the root Swig Win directory to C:\

You should end up with something like this:



Add “C:\swigwin-3.0.12” (or whichever version of SWIG you have downloaded) to PATH, then reboot and the SWIG “installation” is complete.

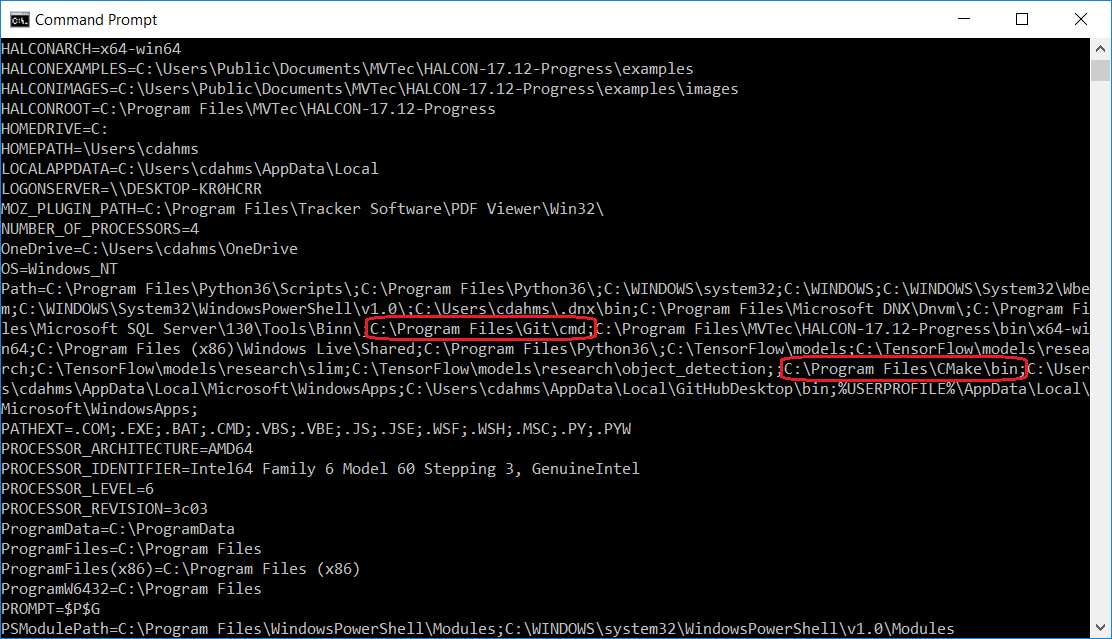
After the reboot, open a command prompt and type “swig --help” to verify your SWIG “install”:



**5) Verify CMake, Git, and SWIG are in your path**

Open a command prompt, type the command “set”, and verify CMake, Git, and SWIG are in your PATH:

**ToDo: update screenshot to include SWIG in PATH**



**6) Run vcvars64.bat**

Visual Studio Community 2015:

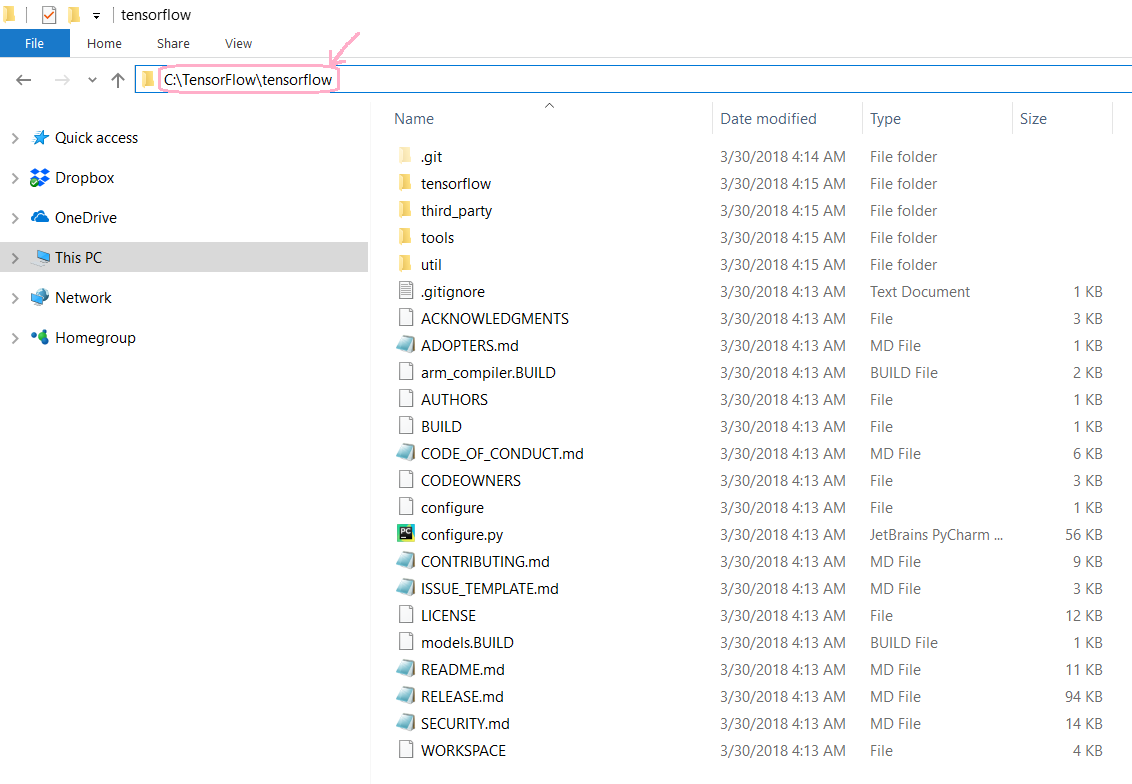
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin\amd64\vcvars64.bat

Visual Studio Community 2017:

C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\VC\Auxiliary\Build\vcvars64.bat

**6) Clone tensorflow to C:\TensorFlow**

Make a directory “C:\TensorFlow” (if you followed the previous tutorials this directory should already exist on your computer), then clone tensorflow <https://github.com/tensorflow/tensorflow> to this location. You should end up with the following:



**6) Clone tensorflow to C:\TensorFlow**

Navigate to “C:\TensorFlow\tensorflow\tensorflow\contrib\cmake” and make a directory “build”, you should now have the path “C:\TensorFlow\tensorflow\tensorflow\contrib\cmake\build” where “build” is an empty directory. Open a command prompt, cd to this empty “build” directory, and enter the following command, all on one line:

cmake .. -A x64

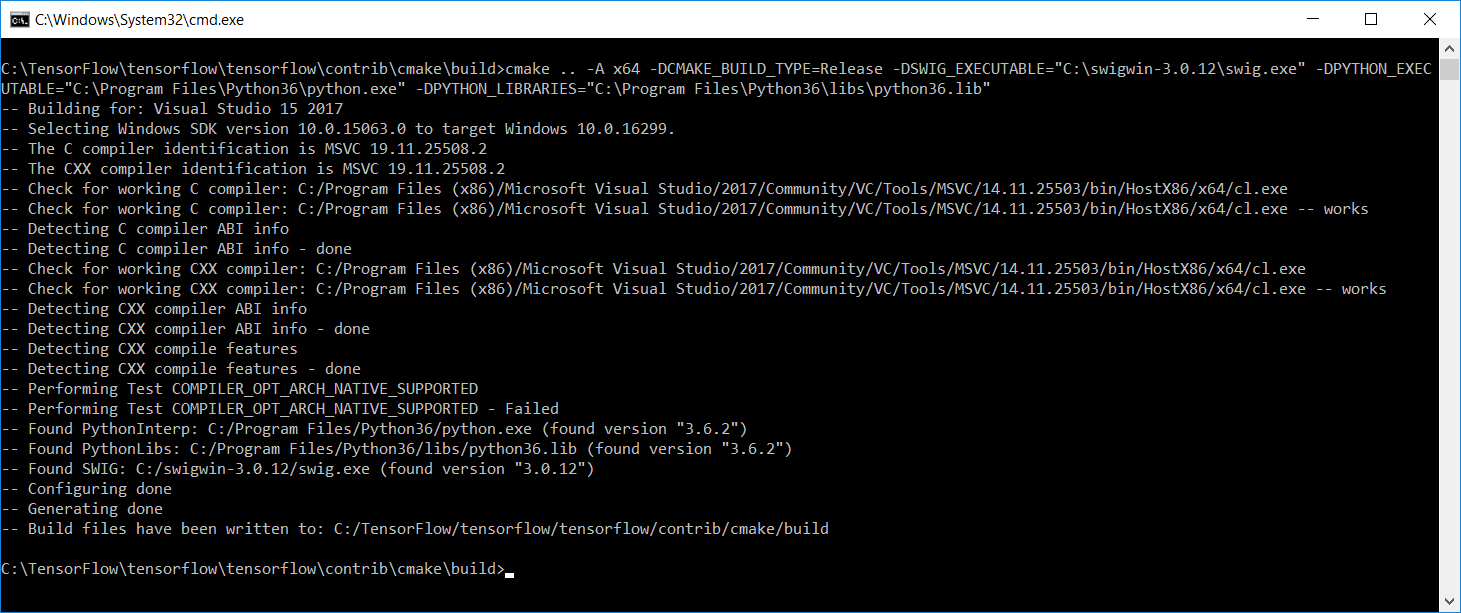
-DCMAKE\_BUILD\_TYPE=Release

-DSWIG\_EXECUTABLE="C:\swigwin-3.0.12\swig.exe"

-DPYTHON\_EXECUTABLE="C:\Program Files\Python36\python.exe"

-DPYTHON\_LIBRARIES="C:\Program Files\Python36\libs\python36.lib"

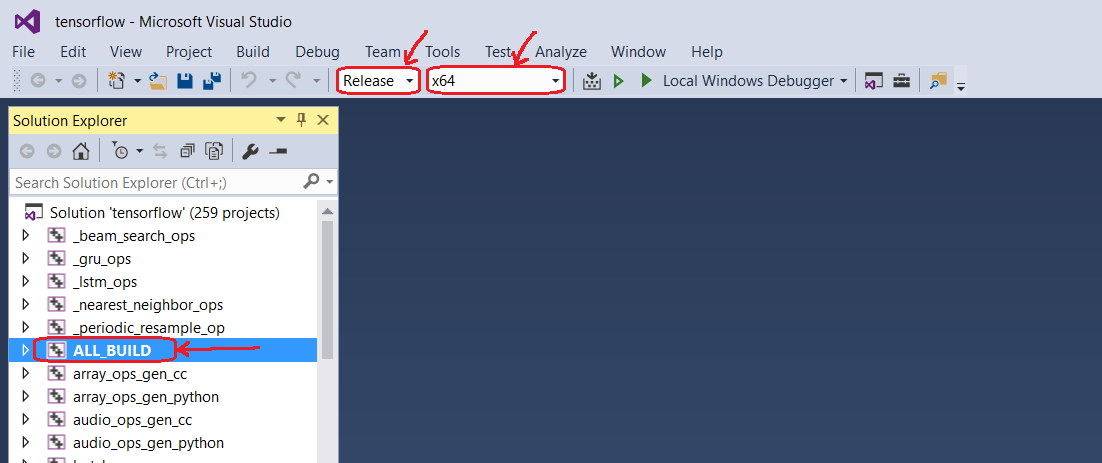
If successful, your command prompt session should look something like this:



**7) kljhasdkjasdjhasd**

The previous step will make many files in “C:\TensorFlow\tensorflow\tensorflow\contrib\cmake\build”, scroll down to “tensorflow.sln” and open it in Visual Studio. This is of course a huge Visual Studio project so it will take a moment to open. Check the progress info at the bottom left of the Visual Studio screen and wait patiently until opening/configuring/parsing/scanning/etc. is complete.

Once open, at the top of Visual Studio, set “Solution Configurations” to “Release” and “Solution Platforms” to “x64”, also verify “ALL\_BUILD” is the default project (should be bold in Solution Explorer):



Now choose Build -> Build. Observe the build getting started for a few minutes and verify there are no errors. Once you’re convinced the build has begun successfully, leave it to complete, which will take perhaps about 2 hours on a typical computer.

**8) kljhasdkjasdjhasd**

**Done!!**

If you’d like to make the accuracy better, the 2 general steps to accomplish this would be:

1) Use more training images. 500-1,000 may seem like a lot, but considering the variety of these images, more would be better. 10,000+ images would not be too many.

2) In retrain.py, set the how\_many\_training\_steps parameter to something higher than 500. Google’s default is 4000. This will make the training take longer, however.

The next tutorial will cover how to use the TensorFlow Object Detection API.