

# **Installation instructions for the tracking-tools repository**

<https://github.com/laxos96/tracking-tools>

**laxos96**

Full installation

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>CUDA and CUDNN</b>	<b>2</b>
<b>3</b>	<b>Tensorflow</b>	<b>3</b>
<b>4</b>	<b>OpenCV</b>	<b>4</b>
<b>5</b>	<b>Visual C++ Build tools</b>	<b>5</b>
<b>6</b>	<b>Darkflow</b>	<b>7</b>
6.1	Possible Errors . . . . .	7
<b>7</b>	<b>mAP</b>	<b>8</b>
<b>8</b>	<b>py-motmetrics</b>	<b>8</b>

## 1 Introduction

This document contains the installation instructions for the tracking-tools repository, aimed at facilitating the task of successfully installing the whole repository.

## 2 CUDA and CUDNN

It is strongly recommended to use the GPU version of tensorflow, so first of all check that your GPU has a CUDA compute capability of at least 3.0. Check here:

| <https://developer.nvidia.com/cuda-gpus>

    If affirmative, proceed to install CUDA following NVIDIA official instructions.

| # Instructions

| [http://developer.download.nvidia.com/compute/cuda/9.0/Prod/docs](http://developer.download.nvidia.com/compute/cuda/9.0/Prod/docs/sidebar/CUDA_Quick_Start_Guide.pdf)  
|     /sidebar/CUDA\_Quick\_Start\_Guide.pdf

| # Download link

| <https://developer.nvidia.com/cuda-90-download-archive>

### **IMPORTANT - INSTALL VERSION 9.0**

    Some versions of tensorflow were not compatible with CUDA versions following 9.0. This version will not give compatibility problems, as far as I know.

    Install CUDNN version 7.0 (again, for best compatibility) from here (remember that you need NVIDIA Developer Program membership, registration is simple).

| <https://developer.nvidia.com/cudnn>

### **IMPORTANT - INSTALL VERSION 7.0**

## 3 Tensorflow

**Remember to activate tensorflow environment, if you are using environments.** If the installation is problematic, check the official guide <https://www.tensorflow.org/install/>, but it should be enough with this command:

```
pip install --upgrade tensorflow-gpu
```

It will ask for a number of dependencies to be installed, among them numpy and matplotlib. We will also install Cython now:

```
pip install cython
```

### Check

In a python console (just write "python" in the cmd with the right environment activated, exit it with "exit()") and execute the following:

```
1 import tensorflow as tf
2 hello = tf.constant('Hello, TensorFlow!')
3 sess = tf.Session()
4 print(sess.run(hello))
```

If tensorflow is imported without errors and the message 'Hello, TensorFlow!' is displayed, installation has been correct. You can also test CUDA now executing the following in a python console:

```
1 import tensorflow as tf
2 tf.test.is_built_with_cuda()
3 tf.test.is_gpu_available()
```

Output should be 'True', and a list of GPU parameters followed by another 'True'. If there is a warning about incompatible versions of tensorflow with numpy or setuptools do not worry, it is a spurious warning that will be fixed in the next tensorflow release and it will execute anyway.

## 4 OpenCV

If you are using conda environments:

```
conda install --channel https://conda.anaconda.org/menpo
  opencv3
```

Otherwise:

```
pip install opencv-python
```

### Check

Open a python console:

```
1 import cv2
2 print(cv2.__version__)
```

Output should be something similar to '3.4.2'. If you have a webcam, you can also test it with a simple script:

```
1 import numpy as np
2 import cv2
3
4 cap = cv2.VideoCapture(0)
5
6 while(True):
7     # Capture frame-by-frame
8     ret, frame = cap.read()
9
10    # Display the resulting frame
11    cv2.imshow('frame',frame)
12    if cv2.waitKey(1) & 0xFF == ord('q'):
13        break
14
15 # When everything done, release the capture
16 cap.release()
17 cv2.destroyAllWindows()
```

## 5 Visual C++ Build tools

Darkflow requires Cython, and therefore C++ build tools are required. These can be downloaded from:

`https://visualstudio.microsoft.com/visual-cpp-build-tools/`

If you have Visual Studio or have previously worked with C++, you probably do not need this step. For the rest, we will download "Build Tools for Visual Studio 2017".

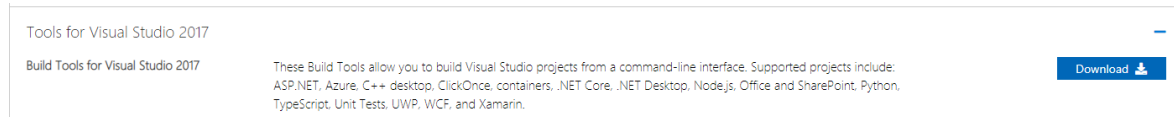


Figure 1

Select the option "Visual C++ build tools".

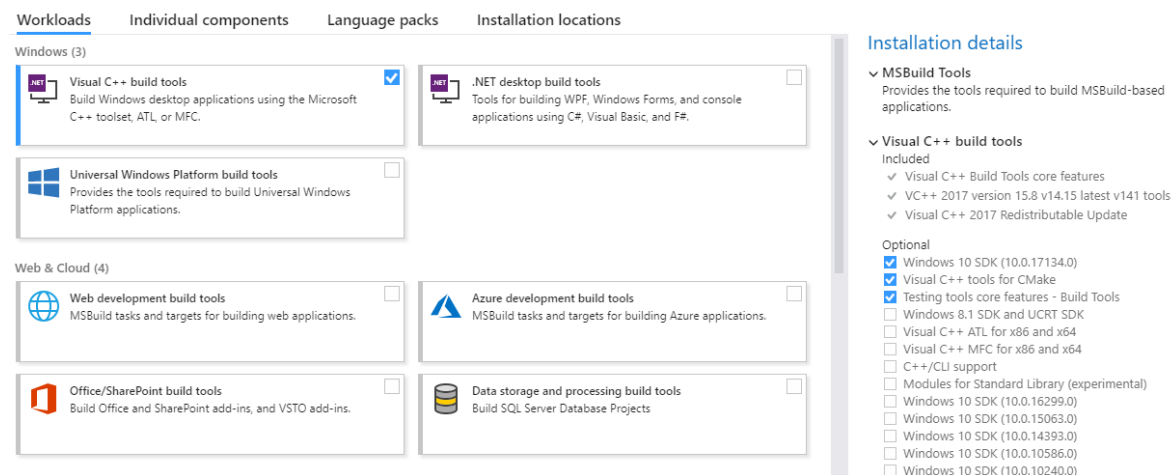
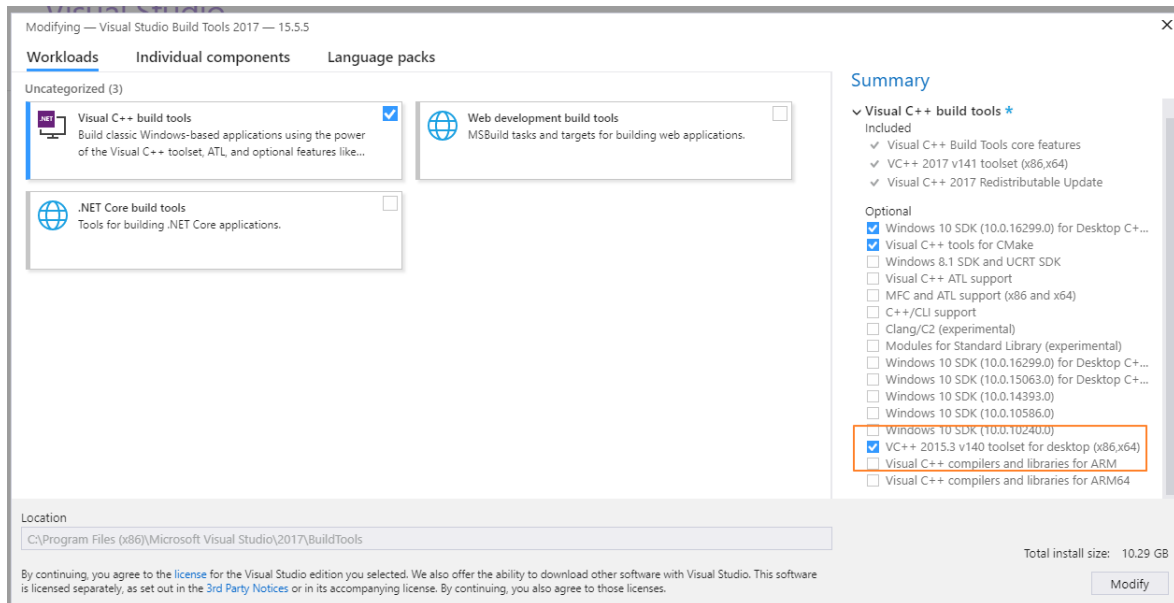


Figure 2

And, in the Individual components section, select the option "VC++ 2015.3 v14.00".

Figure 3: Credit to <https://stackoverflow.com/a/48480367>

## 6 Darkflow

After C++ build tools are installed (up to 10 GB installation), navigate in a cmd to tracking-tools/Tracking-with-darkflow/darkflow. Original creator of this software (<https://github.com/thtrieu/darkflow>) gives three ways of installing this, we will go with the global way. **Remember to activate tensorflow environment, if you are using environments.**

```
pip install -e .
```

### Check

Check that installation has been successful by executing the following command in the tracking-tools/Tracking-with-darkflow/darkflow directory:

```
python flow --h
```

If you see a list of arguments with definitions, everything is OK.

### 6.1 Possible Errors

- If you get this error:

```
error: Microsoft Visual C++ 14.0 is required. Get it with "
Microsoft Visual C++ Build Tools": http://landinghub.
visualstudio.com/visual-cpp-build-tools
```

You may need to upgrade setuptools:

```
pip install --upgrade setuptools
```

- If you get this error:

```
LINK : fatal error LNK1158: cannot run 'rc.exe'
error: command 'D:\\Program Files (x86)\\Microsoft Visual
Studio 14.0\\VC\\BIN\\x86_amd64\\link.exe' failed with
exit status 1158
```

Check this:

```
http://iam-data.com/resolving-python-error-fix-link-fatal-
error-lnk1158-cannot-run-rcexe
```

- If you get this error:

```
darkflow/cython_utils/cy_yolo2_findboxes.pyx: cannot find
  imported module 'nms'
darkflow/cython_utils/cy_yolo2_findboxes.pyx: cannot find
  imported module 'nms'
```

You may be doing the local installation (python setup.py build\_ext -inplace) instead of the recommended global installation, either



```
| pip install -e .  
  
| or  
  
| pip install .
```

## 7 mAP

mAP just requires python and matplotlib, so it should work fine without additional installation.

## 8 py-motmetrics

py-motmetrics requires a simple global installation. **Remember to activate tensor-flow environment, if you are using environments.** Navigate to tracking-tools/py-motmetrics and open a cmd:

```
| pip install .
```

It will automatically install panda and scipy dependencies. Check installation by executing the incorporated test, just type in the cmd:

```
| pytest
```