

How To Train an Hand Detection Classifier Using TensorFlow (GPU) on Windows 10

This tutorial has been created by Valentina Pericu and Sabrina Speranza

1 Installations for Tensorflow GPU 1.9 GPU

We refer here to the instructions provided by Mark Jay: [Tensorflow instructions](#). The prerequisites are: Nvidia GPU, Anaconda with python 3.6, CuDNN v7.1.4 and CUDA v9.0

1.1 CUDA v9.0

Download CUDA v9.0 from [cuda V9.0](#). Choose Windows as Operating System and version 10 and choose `exe(network)`. Cuda requires the installation of Visual Studio and we install Visual Studio professional 2017.

Set your environment variables:

Go to start and search Environment Variables. Click the Environment Variables button. Double click on the Path system variable and select edit. Add the following paths:

- C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v9.0/bin
- C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v9.0/libnvvp
- C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v9.0/extras/CUPTI/libx64

1.1.1 CuDNN v7.1.4

Go to [nvidia account](#) and create your user profile. Install CuDNN v7.1.4 for CUDA v9.0. Extract the zip file contents and place the cuda folder somewhere, for example in C. Add a path to the bin folder so access to the environment variables and in system path add C:/cuda/bin

1.1.2 Update your GPU Driver

Go to [GPU driver install](#), select your GPU version to download and install the driver.

1.1.3 Anaconda with python 3.6

Go to [Anaconda pip 3.6](#) and install python 3.6. Check the "add python to your PATH" option during the installation.

1.2 Install Tensorflow

Open the Anaconda prompt. In C:/users/valen/Anaconda3 create a conda environment, named tensorflow by invoking the following command:

```
conda create -n tensorflow pip python=3.6
```

It says that there exist a new version, so type:

```
conda update -n base conda
```

Activate the conda environment with the following command:

```
conda activate tensorflow
```

Install the GPU version of TensorFlow inside your conda environment by typing the following command:

```
pip install -ignore-installed -upgrade tensorflow-gpu
```

1.2.1 Test your installation

Open the Anaconda prompt and type:

```
python
```

then type:

```
>>>tensorflow as tf
>>>hello = tf.constant('Hello, Tensorflow')
>>>sess = tf.Session()
>>>print(sess.run(hello))
```

the output should be:

```
hello, TensorFlow
```

2 Set up TensorFlow Directory and Anaconda Virtual Environment

2.1 Download TensorFlow Object Detection API repository from GitHub

Create a new folder in C: and name it "tensorflow1". Download the TensorFlow object detection repository from [object detection](#) by clicking the bottom Clone or Download. Open and extract the zip file in folder tensorflow1 and rename "models-master" to "model".

2.2 Download the Faster-RCNN-Inception-V2-COCO model from TensorFlow's model zoo

Download the desired object detection model from [models](#). Here we use the Faster-RCNN-Inception-V2. Download the *faster_rcnn_inception_v2_cocomodel*, open the downloaded *faster_rcnn_inception_v2_model_2018_01_28.tar.gz* file with a file archiver such as WinZip or 7-Zip and extract the *faster_rcnn_inception_v2_coco_2018_01_28* folder to the C:/tensorflow1/models/research/object_detection.

2.3 Download this tutorial's repository from Github

Download the repository located at [repository](#) and extract its content in C : /tensorflow1/models/research/object_detection folder.

To train your own object detector you have to delete the content of the following files:

- all the files in /object_detection/images/train and in /object_detection/images/test
- test_labels.csv and train_labels.csv in /object_detection/images
- all files in /object_detection/training

2.4 Set up new Anaconda virtual environment

First of all run the conda prompt as administrator. Then type the following commands:

- `conda create -n tensorflow1 pip python =3.6`
- `activate tensorflow1`
- `pip install -ignore-installed -upgrade tensorflow-gpu`

After this, you have to install some packages, such as:

- protobuf
- pillow
- lxml
- cython
- jupyter
- scipy
- matplotlib
- pandas
- openCV-python

If the terminal said that Tensorflow 1.9 requires setuptools 39.1, don't worry about that: install your packages with setuptools 39.2 and then substitute it by installing setuptools 39.1 which fits Tensorflow 1.9.

2.5 Configure python path environment

You must create a PYTHONPATH variable that points to the directories: models and research.

To do this, type from any directory:

```
set PYTHONPATH=C:/tensorflow1/models;C:/tensorflow1/models/research;C:/tensorflow1/models/research/slim
```

You have to type this command every time you exit from the virtual environment.

Then, you have to compile Protobufs and run setup.py:

To do this, we follow the instruction provided at [EdgeElectronics tutorial](#), section 2f. Then, from the research directory type the following commands:

```
python setup.py build
```

```
python setup.py install
```

Now you have to check if everything works properly, by typing this from the object_detection directory:

```
jupyter notebook object_detection_tutorial.ipynb
```

This opens the script and allows you to see if code works properly: click run button in the upper toolbar and in the end you should see two labeled images at the bottom section of the page.

3 Dataset

Now we have to download the dataset: we chose the Egohands dataset that you can find here: [Egohands Dataset](#).

Then, put the content of [victordibia/handtracking](#) in the *object_detection* folder.

Now you have to put the *egohands_data.zip* folder inside the *object_detection* one. After that, you must divide the 80% of the dataset images in the train folder, and the 20% in the test folder. In the *egohands_dataset_clean.py* substitute the *dataset_path* with the one of your folder zip *egohands_data.zip*. Then, launch *egohands_dataset_clean.py* which is able to convert the dataset in files csv. Now you should find:

- *test_labels.csv* inside test folder
- *train_labels.csv* inside train folder

Put them out of this folders, inside images.

4 Training

In order to launch the training, follow these steps:

- in the training folder with a text editor create a *labelmap.pbtxt*, our labelmap has only one class. see the link [EdgeElectronics tutorial](#) at section 5 for better instructions.
- from *object_detection/samples/config* take *faster_rcnn_inception_v2.config* and paste it in the training folder.
- you must change this file: see [EdgeElectronics tutorial](#), section 5b.
- Generate the TFRecord files by issuing these commands from the *object_detection* folder:

```
python generate_tfrecord.py -csv_input=images/train_labels.csv -image_dir=images/train
-output_path=train.record
```

```
python generate_tfrecord.py -csv_input=images/test_labels.csv -image_dir=images/test
-output_path=test.record
```

- Run the training from the *object_detection* folder:

```
python train.py -logtostderr -train_dir=training/ -pipeline_config_path=training/faster_rcnn_
```