**In-Depth Windows Tutorial**

Programs needed:

[Anaconda](https://repo.anaconda.com/archive/Anaconda3-2019.10-Windows-x86_64.exe): This is a program that allows you to create multiple environments where you can install packages.

**For GPU use only:**

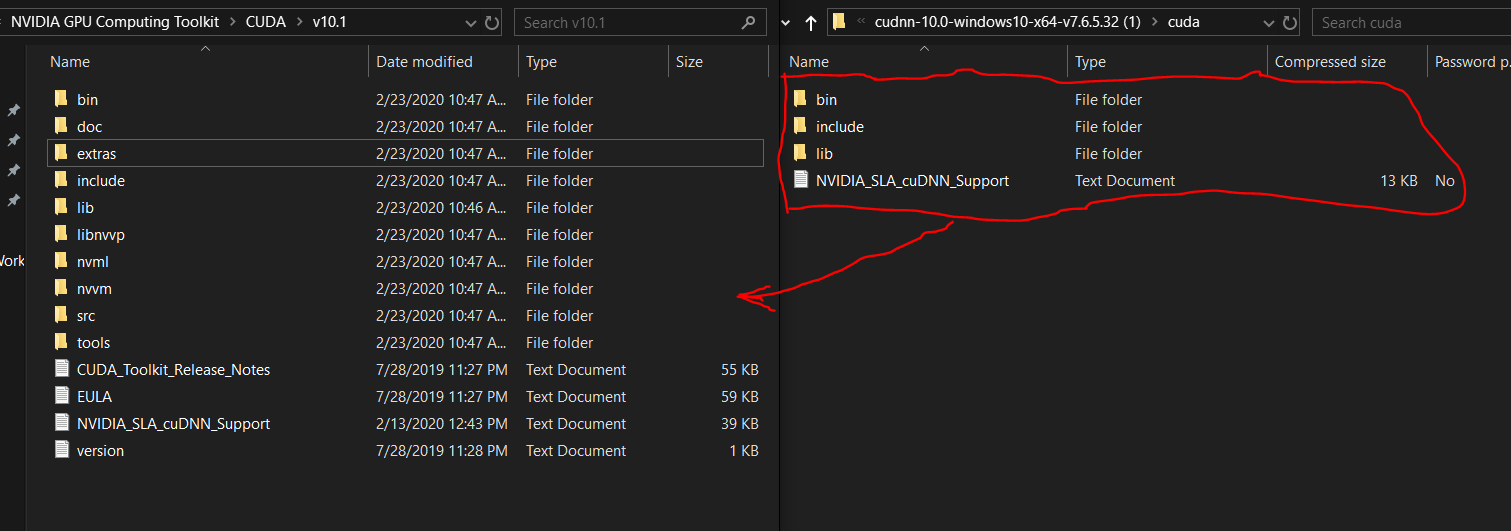
[**CUDA 10.1**](https://developer.nvidia.com/cuda-10.1-download-archive-update2): The drivers needed to run TensorFlow GPU. Currently 10.1 is the most stable version that supports TensorFlow 2.0 For future updates you can use [other versions](https://developer.nvidia.com/cuda-toolkit-archive)

[**CUDNN 7.6.5**](https://developer.nvidia.com/rdp/cudnn-download#a-collapse765-101): The additional files needed for TensorFlow GPU. You will need to create a NVIDIA Developer account. CUDNN 7.6.5 is the most stable version for CUDA 10.1. For future use you can use [other versions](https://developer.nvidia.com/rdp/cudnn-download)

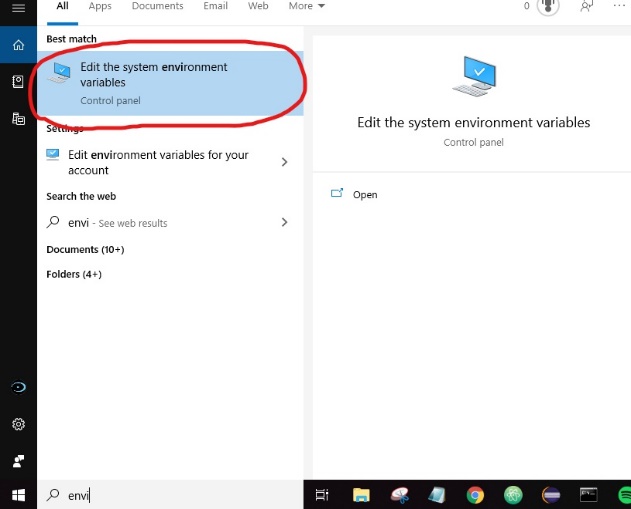
Installation (GPU only):

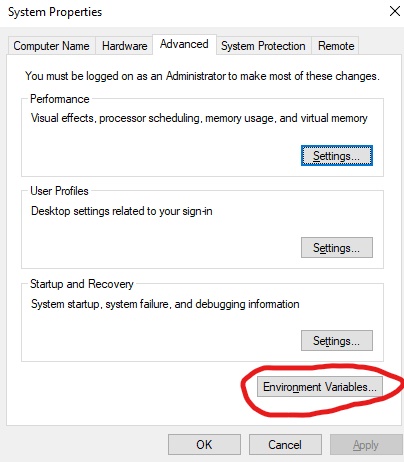
1. Run the CUDA installer
   1. Click OK on the first setup prompt
   2. Click Agree and Continue when prompted about the license agreement
   3. Select the Express installation
   4. Complete final installation steps
2. Add the CUDNN files to CUDA
   1. Open the CUDA path: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1
   2. Open the CUDNN installer, and open the cuda folder
   3. Drag all the contents from the cuda folder to the CUDA path

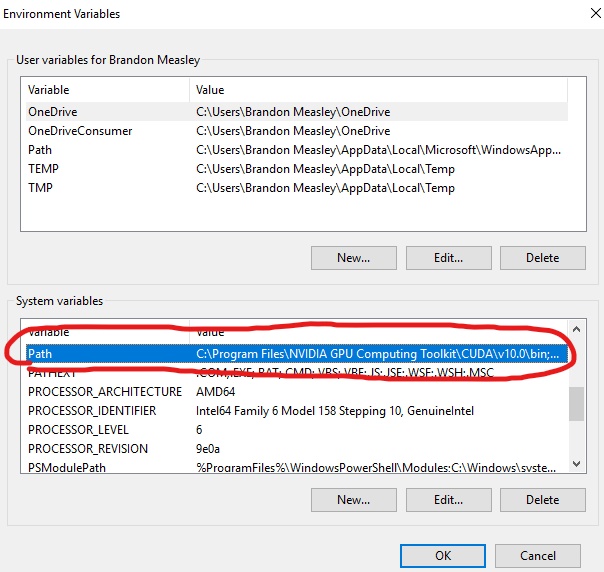
(Click on “Replace all files in folder”)



1. Set up environment variables:
   1. Click on start and type “environment variables”. Select “Edit the system environment variables



* 1. Click on “environment variables” 
  2. Double click on path in the “system variables” section

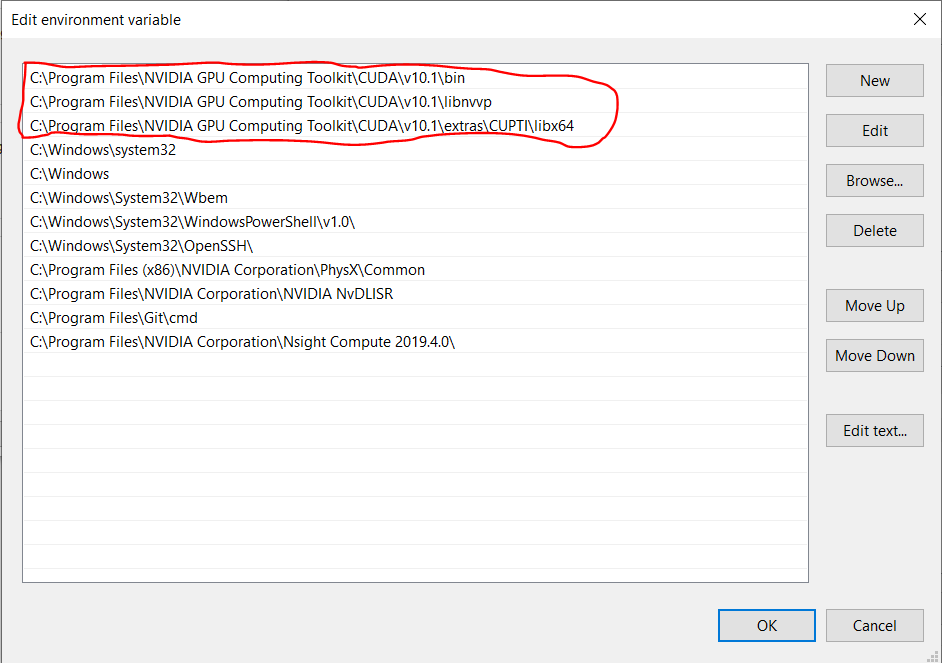


* 1. Enter variables manually by clicking “new”. Make sure you select “ok” when done or the variables will not save. Close the window and reopen to ensure the changes were made. The order of the variables does not matter.

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1\bin

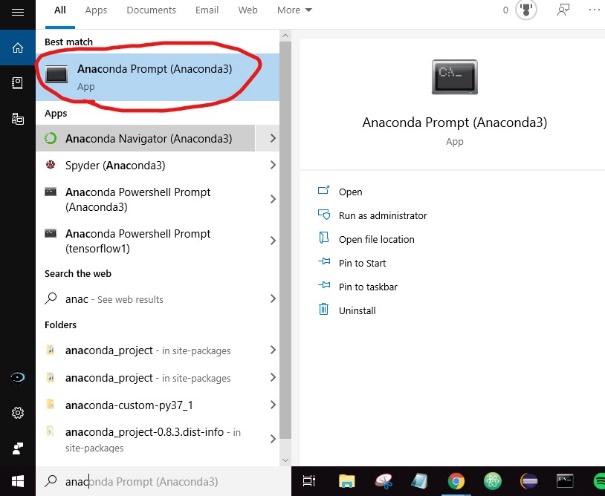
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1\libnvvp

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\extras\CUPTI\libx64



Set-up:

1. Open the conda prompt by clicking on start and typing “anaconda prompt”. Select Anaconda Prompt (Anaconda3)”. This will open a command prompt with in the (base) environment



1. Make sure you have [repository](https://github.com/tylerboice/Digital-Roll) downloaded and unzipped. Change the directory to the location TensorFlow Workbench where you installed the repository using the cd command.

Example if it’s stored on your desktop then run command:

cd Desktop\Digital-Roll\Tensorflow2.0-Workbench

1. Update Conda, run the command:

conda update conda

1. Create a Conda Environment
   1. For CPU use, type the commands:

conda env create -f conda-cpu.yml

conda activate cpu

* 1. For GPU use, type the commands:

conda env create -f conda-gpu.yml

conda activate gpu

1. Install the packages needed for the workbench. Run the command:

pip install -r requirements.txt

Usage:

By default, every time the conda prompt is open you will start in the base environment in home directory. It will look something link:

(base) C:/Users/<your user name>

1. Make sure you are in the environment you desire. To ensure you are in the correct the name to the left on your directory should be the name of your environment ( e.g cpu, gpu ). If you are still in base run:

For CPU use: conda activate cpu

For GPU use: conda activate gpu

Note: to exit an environment type: conda deactivate

1. Find the location where you installed the responsory and change the directory to the Tensorflow2.0-Worbench using the cd command:

Example if it’s stored on your desktop then run command:

cd Desktop\Digital-Roll\Tensorflow2.0-Workbench

1. Before usage insure you have the following files:
   1. The [yolo3.weights](https://pjreddie.com/media/files/yolov3.weights) in your data folder
   2. All images with there corresponding xml file in the images folder. Note: they can be in sub-folders if they are all located in the images folder. Note: if you want to test the workbench on our dataset then you the images found in the “Pre-Trained-Images” folder in the repository
2. Run the workbench

python run\_workbench.py

You should now be in the workbench (<WORKBENCH>). The workbench has been created to be as user friendly as possible. Enter help or h to l