Installation Mannual for Object Deetction

Getch

March 2024

1 Installation

The first activity is creating a conda environment. To create a conda environment on the Windows operating system, stand-alone Anaconda or Miniconda should be installed. package and installation instructions can be accessed from `https://www.anaconda.com/`. In addition to this, it is highly recommended to have Microsoft Visual C^{++} build tools (see `https://visualstudio.microsoft.com/downloads/?q=build+tools`) in the system as it is a dependency tool for many Python packages. For a machine that already has a functioning Cuda installation, this could be skipped. Then start the anaconda command line interface by typing anaconda in the Windows search which gives the anaconda command line interface. Then create and activate the new anaconda environment by typing the following command.

```
conda create —n my_env
conda activate my_env
```

Please note that **my_env** could be any environment name but should be without spaces

In the code snipped above, **my_env** could be any name for our new conda environment. Concerning the creation of a conda environment, it is also possible to have a virtual environment but for effective package and version conflict management, anaconda is mostly recommended.

Once the environment is created, the next task is installing packages. Here it should be noted that there are some base packages automatically installed with the new conda environment. The first package we need to install is packages dedicated to deep learning framework, PyTorch and Torchvision.

Packages that did not need special requirements related to physical hardware infrastructure like operating system and CUDA[1] versions(which are also synchronized with available GPU/CPU resources) are provided in a text document in the requirements text document. After locating this requirements document, these packages could be installed using the following script.

[1] The installation of operating system specific cuda toolkit could be accessed from `https://developer.nvidia.com/cuda-downloads?target_os=Windows`

```
cd /MultiMaskRCNN # This should be to a folder where the requirements.txt file is located

pip install -r requirements.txt
```

Installing deep learning packages like Pytorch requires careful specification of the operating system, python version, and installed Cuda version. Default installation procedures for the latest Pytorch and Cuda versions are available at https://pytorch.org/get-started/locally/.

If our machine is equipped with Python and GPU Cuda that are not of the latest versions, we can alternatively select specific Pytorch and Torchvision package versions from https://pytorch.org/get-started/previous-versions/. For example, using the following command to install PyTorch compatible with CUDA version 11.6

```
conda install pytorch==1.13.1 torchvision==0.14.1 pytorch-cuda=11.6 -c pytorch -c nvidia
```

Another package that is very important for object detection dataset preparation and evaluation of model prediction is **COCO API**(https://github.com/cocodataset/cocoapi/tree/master), which is released together with MS COCO dataset(https://cocodataset.org/#home). It is also used in most object detection tasks. Installation of COCO API is very straightforward in the Linux operating system. In Windows operating systems, being the installation is tricky, the open-source community has devised customized versions. For example, installing the package from the https://github.com/philferriere/cocoapi using the command:

```
pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI
```

There are also alternatives recommended to install cocoapi on linux operating system.

```
pip install pycocotools
```

One of the packages that is mostly used as a utility to properly read geospatial images like geotif images is **imagecodecs** package. It can be installed using the following command;

```
pip install imagecodecs
```

Please note that sometimes it pops up with error messages. It is highly recommended to check the official installation and dependency issues at https://pypi.org/project/imagecodecs/.

2 Checking installed packages

Once all necessary packages are installed the following checks can be run. After running the anaconda prompt type the following and press enter key:

```
python -c "import torch; import torchvision"
```

if there is no any error message, then the installation is successfull.

Alternatively, this could be done by typing

```
python
```

on the anaconda command line which leads to the python command line, and then import the packages

```
import torch # to check the pytorch installation
import torchvision # to check the torchvision installation
import coco # to check coco installation
```

Note: *Please note that python command line interface its recommended to run and check each line alone to make sure all necessary packages are installed.* Then make sure to exit the Python command line interface and be on the conda command line interface.

for inspection of pytorch models installing package **torchsummary** using the following command as:

```
python –m pip install torchsummary
```

3 Inspecting the GPU availability and model size

Once we are sure the necessary packages are installed and are working properly, the next task is running our script. Before that, we have to make sure we have GPU resources and the intended model can fit on it.

To inspect the availability of GPU resources, a path to the folder where the **”inspection.py”** file is saved. Then run the following command which provides a summary of the availability of the GPU, its size, and proposed model size.

```
python inspection.py
```