



## Installing Keras Mask R-CNN

The instructions to install Keras Mask R-CNN are similar to our [install instructions for Keras RetinaNet](#).

Throughout this book I have recommended using Python virtual environments to keep our development environments separate from each other. I would therefore recommend you do the same, *although* I will say that your virtual environment used for Mask R-CNN will be near identical as your RetinaNet one so you could *technically* re-use the same environment (although I don't recommend it).

I'll make the assumption you'll be creating a new Python virtual environment for Mask R-CNN — let's go ahead and do that now:

```
Installing Keras Mask R-CNN Shell
1 $ mkvirtualenv mask_rcnn -p python3
```

Next, we need to ensure the following Python packages are installed:

```
Installing Keras Mask R-CNN Shell
1 $ workon mask_rcnn
2 $ pip install numpy scipy h5py
3 $ pip install scikit-learn Pillow
4 $ pip install imgaug imutils
5 $ pip install beautifulsoup4
6 $ pip install tensorflow-gpu==1.12
7 $ pip install keras
```

Note that TensorFlow is a *requirement* when utilizing Keras + Mask R-CNN. Here I am using the GPU version of TensorFlow (`tensorflow-gpu`). You could install the CPU-only version of TensorFlow (`tensorflow`); however, there is an incredible amount of computation performed in a Mask R-CNN

and in most cases it will be entirely intractable to train a Mask R-CNN on your CPU.

If you run into any issues installing TensorFlow, make sure you refer to the [official TensorFlow install documentation](#).

You'll also want to sym-link your OpenCV `cv2` bindings into your Python virtual environment (provided you are using one):

```
Installing Keras Mask R-CNN Shell
1 $ cd ~/.virtualenvs/mask_rcnn/lib/python3.4/site-packages/
2 $ ln -s /usr/local/lib/python3.4/site-packages/cv2.cpython-35m-x86_64-linux-gnu.so cv2.so
```

Your path to your OpenCV bindings will be different than mine so make sure you check your system before executing the commands above. Make sure you:

- Refer to the development environment configuration instructions on the [companion website homepage](#).
- Verify your OpenCV install path before creating the sym-link (otherwise the sym-link will point to a nonexistent file and the import will fail).

If you are having problems with your OpenCV install you may want to try a simple pip install of the library:

```
Installing Keras Mask R-CNN Shell
1 $ cd ~
2 $ pip install opencv-contrib-python
```

I provide more information on installing OpenCV via pip in [this blog post](#).

The nextstep is to clone down the Keras + Mask R-CNN implementation for [their official GitHub project page](#):

```
Installing Keras Mask R-CNN Shell
1 $ cd ~
2 $ git clone https://github.com/matterport/Mask_RCNN
3 $ cd Mask_RCNN
4 $ git checkout 1aca439c37849dcd085167c4e69d3abcd9d368d7
```

We'll be using the [v2.1 release](#) as our base; however, keep in mind that the Keras + Mask R-CNN implementation is under *active development* so make sure you checkout the *same* version of the Mask R-CNN library as I am using.

Next, install all requirements for the Mask R-CNN package:

```
Installing Keras Mask R-CNN Shell
1 $ pip install -r requirements.txt
```

To verify that the Keras + Mask R-CNN package is successfully installed, from the `Mask_RCNN` directory, open up a Python shell and try to import the `mrcnn` package:

```
Installing Keras Mask R-CNN Shell
1 $ python
2 >>> import mrcnn
3 >>>
```

If the import is successful with no errors then `mrcnn` is properly installed.

**Note:** We'll be sym-linking the `mrcnn` directory into our working project directory so we don't need to run the `setup.py` script.

At this point your Keras + Mask R-CNN install is complete; however, in order to run the examples in the text you'll want to download the `mask_rcnn_coco.h5` file from the official [Mask R-CNN release page](#). The direct link to the model file can be found [here](#).

This model has been pre-trained on the COCO dataset and will serve as our starting point. We'll be fine-tuning this model for instance segmentation on our own dataset. **Make sure you download the file now and keep a local copy that you can when following along with the text!**

## Frequently Asked Questions (FAQ)

This section provides answers to frequently asked questions or problems when using the Keras implementation of Mask R-CNN.

**Q.** I am trying to train my Mask R-CNN with my GPU; however, only my CPU is being used. My GPU is not being used for training. Why is that?

**A.** Be sure to check your output of `pip freeze`. You likely have both `tensorflow` and `tensorflow-gpu` installed. If both are installed then TensorFlow will default to using the standard tensorflow (CPU) library.

The solution is to uninstall both `tensorflow` and `tensorflow-gpu` then install *just* `tensorflow-gpu`:

```
Installing TensorFlow GPU Shell
1 $ pip uninstall tensorflow
2 $ pip uninstall tensorflow-gpu
3 $ pip install tensorflow-gpu
```

I believe this issue is caused by a bug in the `requirements.txt` file for Keras Mask R-CNN. Inside the `requirements.txt` file the CPU version is installed; however, this causes problems if the GPU version of TensorFlow is *already* installed. The above solution resolves the issue.

## Links

---

- [Supplementary material](#)
- [Bug tracking and issues](#)
- [PyImageSearch contact form](#)

---

© 2020 PyImageSearch. All Rights Reserved.