1. The first step is to load a fast R-CNN network in the “fast\_rcnn\_load\_net.m”.

Run:./data/scripts/fetch\_fast\_rcnn\_models.sh

1. After that, to train and test with PASCAL VOC, you will need to establish symlinks.

From the `data` directory (`cd data`):

For VOC 2007

ln -s /your/path/to/VOC2007/VOCdevkit VOCdevkit2007

Since you'll likely be experimenting with multiple installs of Fast R-CNN in

parallel, you'll probably want to keep all of this data in a shared place and

use symlinks. On my system I create the following symlinks inside `data`:

```

# data/cache holds various outputs created by the datasets package

ln -s /data/fast\_rcnn\_shared/cache

# move the imagenet\_models to shared location and symlink to them

ln -s /data/fast\_rcnn\_shared/imagenet\_models

# move the selective search data to a shared location and symlink to them

ln -s /data/fast\_rcnn\_shared/selective\_search\_data

ln -s /data/VOC2007/VOCdevkit VOCdevkit2007

ln -s /data/VOC2012/VOCdevkit VOCdevkit2012

1. After he first step, we have load the Fast R-CNN as well as environment to store data.
2. The next step is to perform detection a Fast R-CNN network given an image and object proposals.
3. For the tested and trained output:

Artifacts generated by the scripts in `tools` are written in this directory.

* Trained Fast R-CNN networks are saved under:

```

output/<experiment directory>/<dataset name>/

```

* Test outputs are saved under:

```

output/<experiment directory>/<dataset name>/<network snapshot name>/

```