
(http://lib.csdn.net/base/deeplearning)

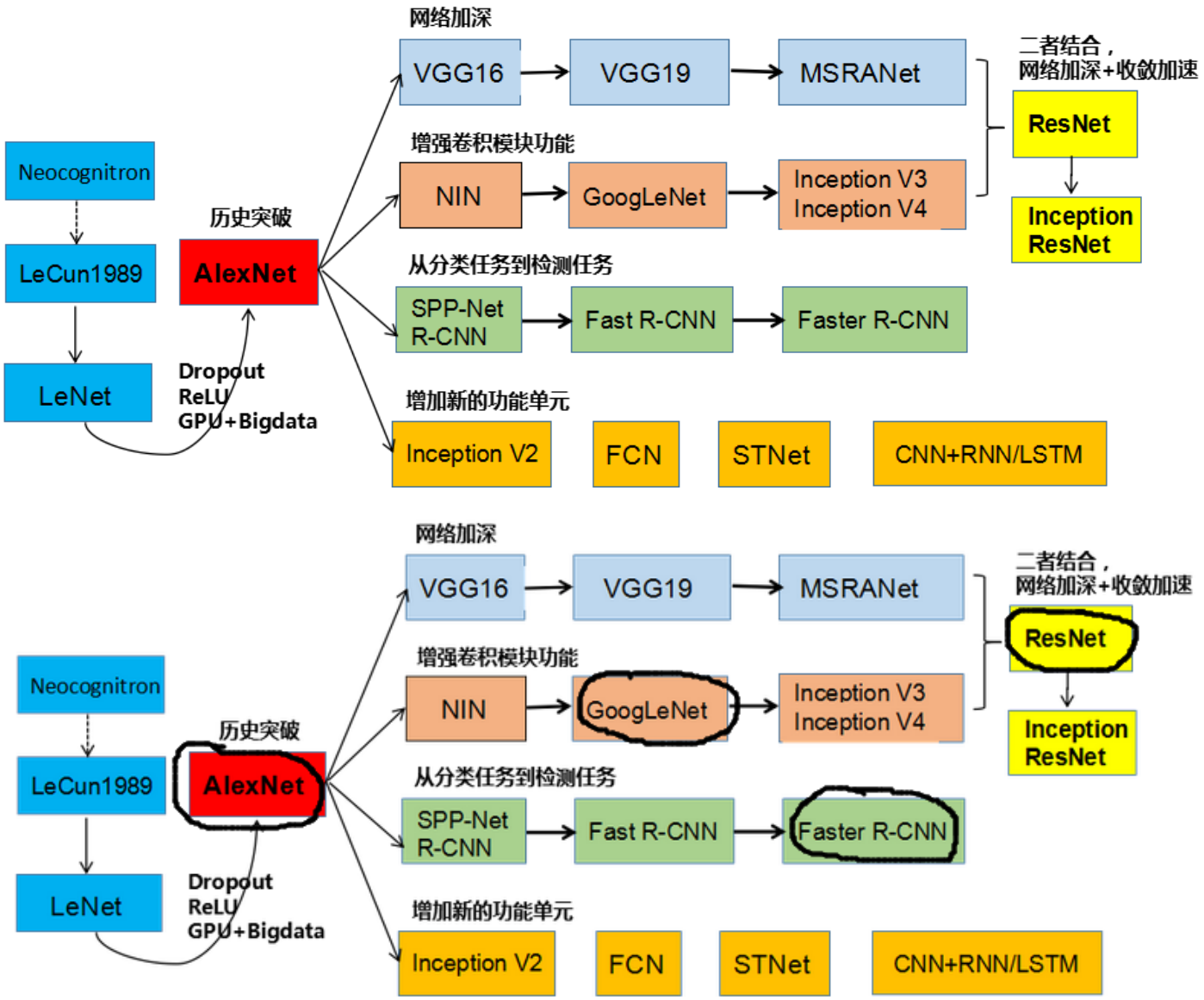
深度学习 (http://lib.csdn.net/base/deeplearning) - 深度学习应用 (http://lib.csdn.net/deeplearning/node/747) - 图像检测 (http://lib.csdn.net/deeplearning/knowledge/1726)

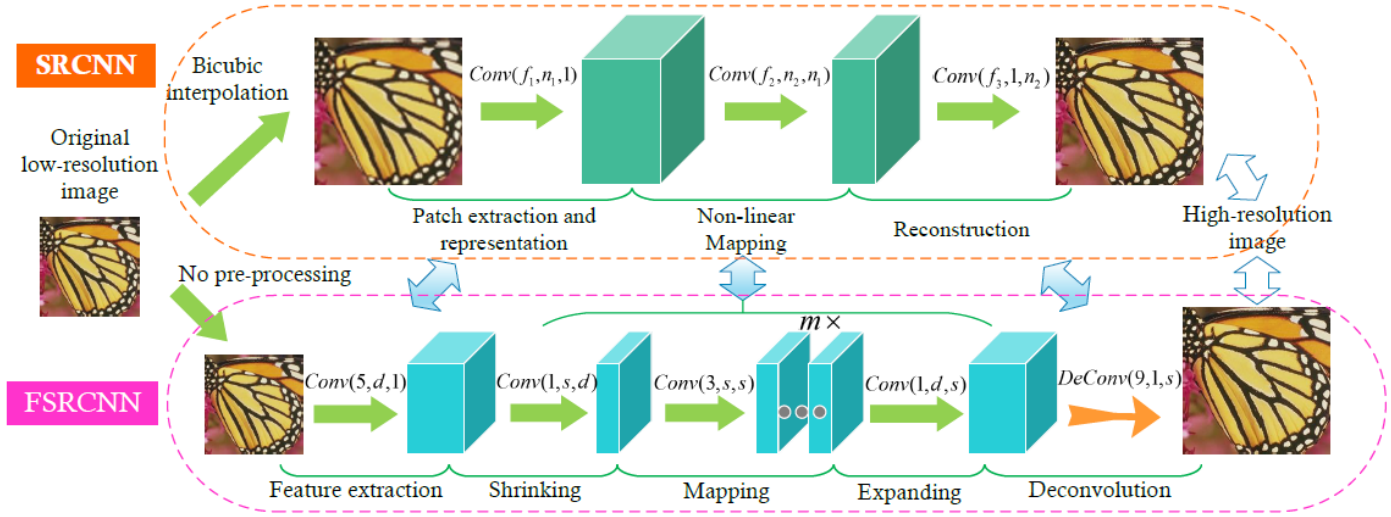
caffe-fast-rcnn（Caffe、FSRCNN、FastRCNN）

作者：forest_world (http://my.csdn.net/forest_world)

转载请注明：http://blog.csdn.net/forest_world (http://blog.csdn.net/forest_world)

一、文件架构


网络加深
VGG16
VGG19
MSRANet
二者结合，
网络加深+收敛加速
ResNet
Inception
ResNet
增强卷积模块功能
NIN
GoogLeNet
Inception V3
Inception V4
从分类任务到检测任务
SPP-Net
R-CNN
Fast R-CNN
Faster R-CNN
增加新的功能单元
Inception V2
FCN
STNet
CNN+RNN/LSTM
Neocognitron
LeCun1989
LeNet
历史突破
AlexNet
Dropout
ReLU
GPU+Bigdata



二、FSRCNN开发环境搭建：

```
faster-rcnn:
matlab版本ShaoqingRen/faster_rcnn: Faster R-CNN
rbg提供的python版本rbgirshick/py-faster-rcnn
```

```
git clone https://github.com/LMDB/lmdb
Cloning into 'lmdb'...
remote: Counting objects: 7201, done.
remote: Total 7201 (delta 0), reused 0 (delta 0), pack-reused 7201
Receiving objects: 100% (7201/7201), 1.40 MiB | 7.00 KiB/s, done.
Resolving deltas: 100% (3097/3097), done.
Checking connectivity... done.
```

安装lmdb

cd liblmdb

make

sudo make install

git clone –recursive https://github.com/rbgirshick/fast-rcnn.git (https://github.com/rbgirshick/fast-rcnn.git)

编译：Cython module

cd lib

make

编译：Caffe and pycaffe

cd caffe-fast-rcnn

make -j8 && make pycaffe

```
@ubuntu:~/fast-rcnn/caffe-fast-rcnn$ make -j8 && make pycaffe
LD -o .build_release/lib/libcaffe.so
CXX/LD -o .build_release/tools/compute_image_mean.bin
CXX/LD -o .build_release/tools/upgrade_net_proto_binary.bin
CXX/LD -o .build_release/tools/upgrade_net_proto_text.bin
CXX/LD -o .build_release/tools/finetune_net.bin
CXX/LD -o .build_release/tools/net_speed_benchmark.bin
CXX/LD -o .build_release/tools/train_net.bin
CXX/LD -o .build_release/tools/caffe.bin
CXX/LD -o .build_release/tools/convert_imageset.bin
CXX/LD -o .build_release/tools/extract_features.bin
CXX/LD -o .build_release/tools/device_query.bin
CXX/LD -o .build_release/tools/test_net.bin
CXX/LD -o .build_release/examples/cifar10/convert_cifar_data.bin
CXX/LD -o .build_release/examples/mnist/convert_mnist_data.bin
CXX/LD -o .build_release/examples/siamese/convert_mnist_siamese_data.bin
CXX/LD -o python/caffe/_caffe.so python/caffe/_caffe.cpp
touch python/caffe/proto/__init__.py
PROTOC (python) src/caffe/proto/caffe.proto
```

错误1：

```
LD -o .build_release/lib/libcaffe.so
.build_release/src/caffe/layers/absval_layer.o: file not recognized: File truncate
collect2: error: ld returned 1 exit status
make: *** [.build_release/lib/libcaffe.so] Error 1
```

```
CXX src/caffe/layers/dummy_data_layer.cpp
In file included from ./include/caffe/layer.hpp:8:0,
                 from src/caffe/layers/relu_layer.cpp:4:
./include/caffe/blob.hpp:9:34: fatal error: caffe/proto/caffe.pb.h: No such file o
 #include "caffe/proto/caffe.pb.h"
                                  ^
compilation terminated.
The bug is not reproducible, so it is likely a hardware or OS problem.
make: *** [.build_release/src/caffe/layers/relu_layer.o] Error 1
make: *** Waiting for unfinished jobs....
In file included from ./include/caffe/fast_rcnn_layers.hpp:13:0,
                 from src/caffe/layers/roi_pooling_layer.cpp:10:
./include/caffe/blob.hpp:9:34: fatal error: caffe/proto/caffe.pb.h: No such file o
 #include "caffe/proto/caffe.pb.h"
                                  ^
compilation terminated.
The bug is not reproducible, so it is likely a hardware or OS problem.
make: *** [.build_release/src/caffe/layers/roi_pooling_layer.o] Error 1
```

解决：

```
~/fast-rcnn/caffe-fast-rcnn/src/caffe/proto$ protoc --cpp_out=/home/hello/fast-rcnn/caffe-fas
```

sudo apt-get install libatlas-base-dev

三、训练检测

下载模型文件：

```
@ubuntu:~/fast-rcnn$ ./data/scripts/fetch_fast_rcnn_models.sh
Downloading Fast R-CNN demo models (0.96G)...
--2016-11-08 11:01:15--  http://www.cs.berkeley.edu/~rbg/fast-rcnn-data/fast_rcnn_
Resolving www.cs.berkeley.edu (www.cs.berkeley.edu)... 23.253.180.102
Connecting to www.cs.berkeley.edu (www.cs.berkeley.edu)|23.253.180.102|:80... conn
HTTP request sent, awaiting response... 302 Found
Location: http://101.96.10.61/www.cs.berkeley.edu/~rbg/fast-rcnn-data/fast_rcnn_mo
--2016-11-08 11:01:16--  http://101.96.10.61/www.cs.berkeley.edu/~rbg/fast-rcnn-da
Connecting to 101.96.10.61:80... connected.
HTTP request sent, awaiting response... 303 See Other
Location: https://people.eecs.berkeley.edu/~rbg/fast-rcnn-data/fast_rcnn_models.tg
```

1、出现问题：

```
~/fast-rcnn/tools$ ./demo.py
Traceback (most recent call last):
  File "./demo.py", line 17, in <module>
    from fast_rcnn.config import cfg
  File "/home//fast-rcnn/tools/../lib/fast_rcnn/__init__.py", line 8, in <module>
    from . import config
  File "/home//fast-rcnn/tools/../lib/fast_rcnn/config.py", line 23, in <module>
    from easydict import EasyDict as edict
ImportError: No module named easydict
```

解决：

sudo apt-get install python-pip

sudo pip install easydict

```
~/fast-rcnn/tools$ ./demo.py
Traceback (most recent call last):
  File "./demo.py", line 17, in <module>
    from fast_rcnn.config import cfg
  File "/home//fast-rcnn/tools/../lib/fast_rcnn/__init__.py", line 9, in <module>
    from . import train
  File "/home//fast-rcnn/tools/../lib/fast_rcnn/train.py", line 10, in <module>
    import caffe
  File "/home//fast-rcnn/tools/../caffe-fast-rcnn/python/caffe/__init__.py", line
    from .pycaffe import Net, SGDSolver
  File "/home//fast-rcnn/tools/../caffe-fast-rcnn/python/caffe/pycaffe.py", line 1
    import caffe.io
  File "/home//fast-rcnn/tools/../caffe-fast-rcnn/python/caffe/io.py", line 2, in
    import skimage.io
ImportError: No module named skimage.io
```

sudo pip install scikit-image

sudo apt-get install python-numpy python-scipy python-matplotlib python-sklearn python-skimage python-h5py python-protobuf python-leveldb python-networkx python-nose python-pandas python-gflags Cython ipython

2、出现问题：

```
@ubuntu:~/fast-rcnn/tools$ ./demo.py
WARNING: Logging before InitGoogleLogging() is written to STDERR
F1108 15:18:01.710467 13445 common.cpp:55] Cannot use GPU in CPU-only Caffe: check
*** Check failure stack trace: ***
Aborted (core dumped)
```

解决：

```
#if args.cpu_mode:
        caffe.set_mode_cpu()
    #else:
        #caffe.set_mode_gpu()
        #caffe.set_device(args.gpu_id)
```

3、出现问题：

```
(or to disable these warnings), see CodedInputStream::SetTotalBytesLimit() in google/prot
```

解决：

内存问题

测试结果：

```
Loaded network /home//fast-rcnn/data/fast_rcnn_models/vgg16_fast_rcnn_iter_40000.c
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Demo for data/demo/000004.jpg
Detection took 56.309s for 2888 object proposals
All car detections with p(car | box) >= 0.8
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Demo for data/demo/001551.jpg
Detection took 40.754s for 2057 object proposals
All sofa detections with p(sofa | box) >= 0.8
All tvmonitor detections with p(tvmonitor | box) >= 0.8
```

参考资料：

1、http://blog.csdn.net/cyh_24/article/details/51440344 (http://blog.csdn.net/cyh_24/article/details/51440344) 卷积神经网络-进化史 从LeNet到AlexNet

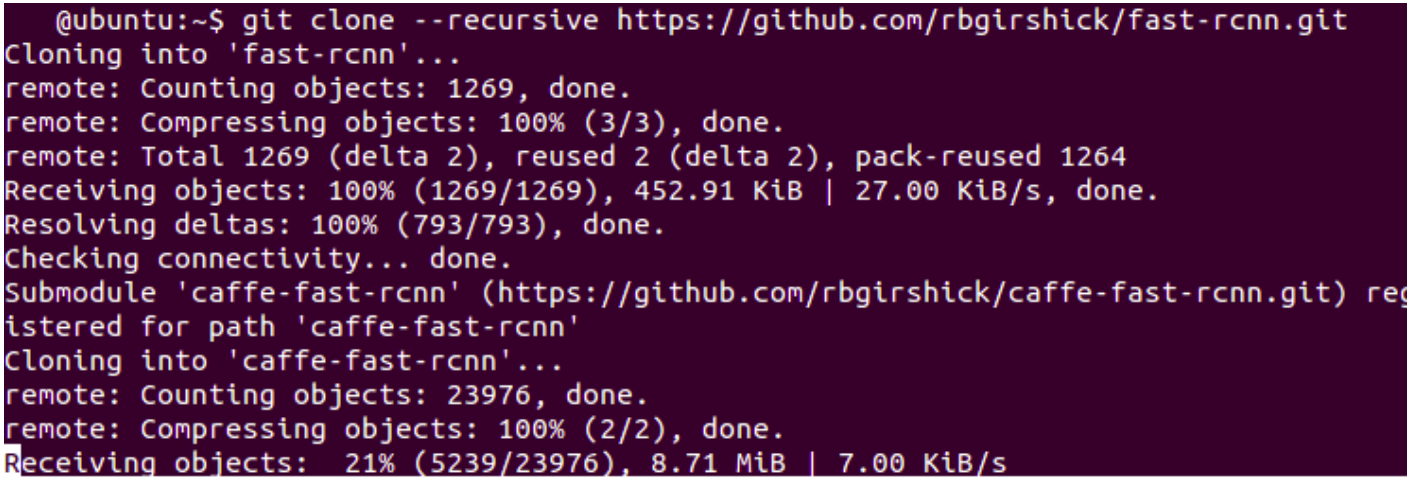
2、Accelerating the Super-Resolution Convolutional Neural Network

Chao Dong, Chen Change Loy, and Xiaoou Tang

Department of Information Engineering, The Chinese University of Hong Kong

3、 http://www.cnblogs.com/louyihang-loves-baiyan/p/4885659.html (http://www.cnblogs.com/louyihang-loves-baiyan/p/4885659.html) FastRCNN 训练自己数据集 (1编译配置)

git clone –recursive https://github.com/rbgirshick/fast-rcnn.git (https://github.com/rbgirshick/fast-rcnn.git)



```
ubuntu:~$ git clone --recursive https://github.com/rbgirshick/fast-rcnn.git
fatal: destination path 'fast-rcnn' already exists and is not an empty directory.
@ubuntu:~$ rm -rf fast-rcnn/
@ubuntu:~$ git clone --recursive https://github.com/rbgirshick/fast-rcnn.git
Cloning into 'fast-rcnn'...
remote: Counting objects: 1269, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 1269 (delta 2), reused 2 (delta 2), pack-reused 1264
Receiving objects: 100% (1269/1269), 452.91 KiB | 27.00 KiB/s, done.
Resolving deltas: 100% (793/793), done.
Checking connectivity... done.
Submodule 'caffe-fast-rcnn' (https://github.com/rbgirshick/caffe-fast-rcnn.git) re
Cloning into 'caffe-fast-rcnn'...
remote: Counting objects: 23976, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 23976 (delta 0), reused 0 (delta 0), pack-reused 23974
Receiving objects: 100% (23976/23976), 31.60 MiB | 37.00 KiB/s, done.
Resolving deltas: 100% (15681/15681), done.
Checking connectivity... done.
Submodule path 'caffe-fast-rcnn': checked out 'bcd9b4eadc7d8fbc433aeefd564e82ec63a
```

出现问题:

```
@ubuntu:~/fast-rcnn/lib$ make
python setup.py build_ext --inplace
Traceback (most recent call last):
  File "setup.py", line 11, in <module>
    from Cython.Distutils import build_ext
ImportError: No module named Cython.Distutils
make: *** [all] Error 1
```

解决：sudo apt-get install cython

sudo apt-get install protobuf-devel leveldb-devel snappy-devel opencv-devel boost-devel hdf5-devel

4、http://www.w2bc.com/article/125121 (http://www.w2bc.com/article/125121) 目标检测 Faster RCNN算法详解

5、http://www.w2bc.com/article/168733 (http://www.w2bc.com/article/168733) rcnn fast-rcnn faster-rcnn资料

6、http://www.w2bc.com/article/128354 (http://www.w2bc.com/article/128354) RCNN学习笔记(7):Faster R-CNN 英文论文翻译笔记

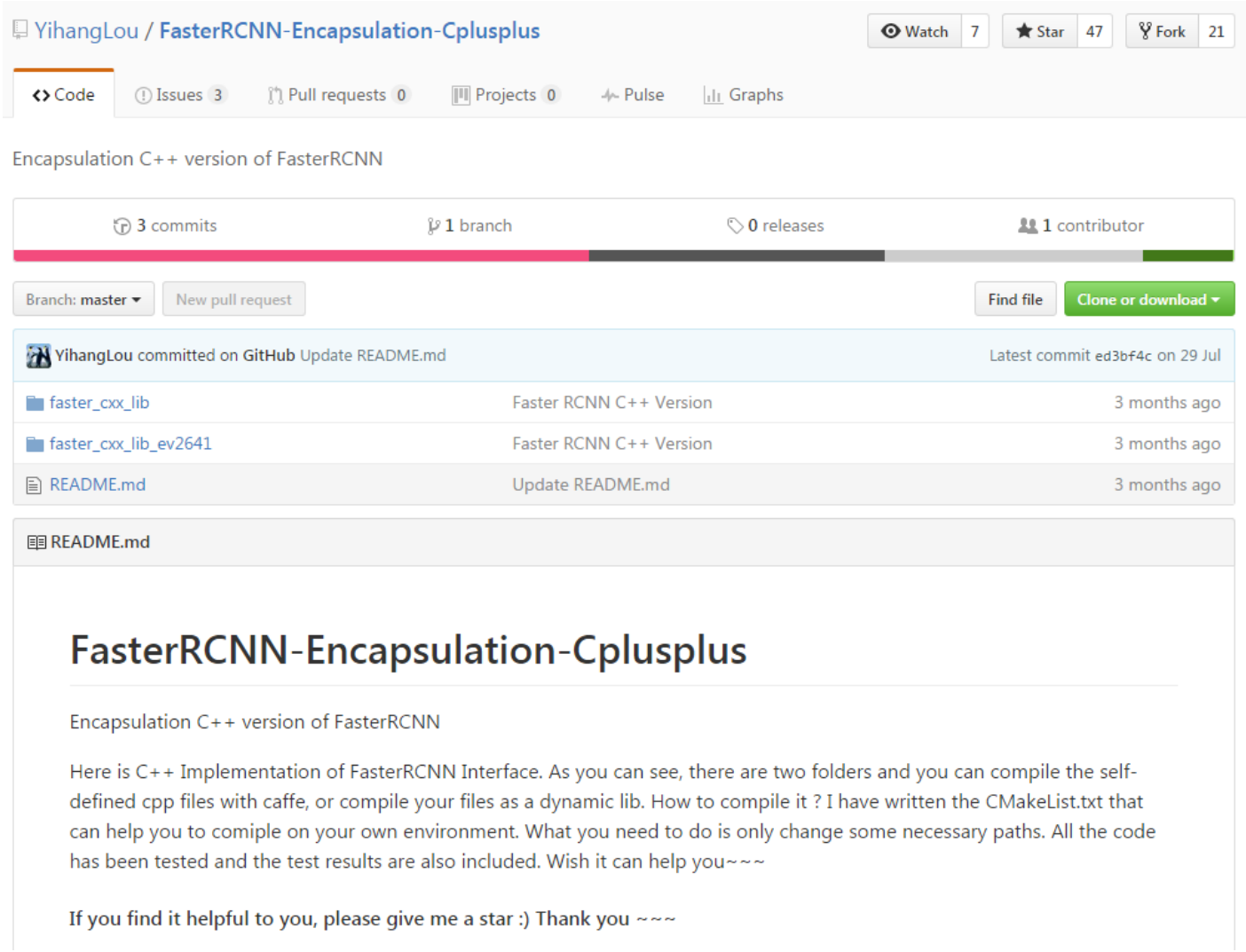
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

7、http://www.w2bc.com/article/120530 (http://www.w2bc.com/article/120530) 目标检测 Fast RCNN算法详解

8、http://www.w2bc.com/article/136766 (http://www.w2bc.com/article/136766) faster_rcnn c++版本的 caffe 封装，动态库（2）

9、http://www.cnblogs.com/louyihang-loves-baiyan/p/5485955.html (http://www.cnblogs.com/louyihang-loves-baiyan/p/5485955.html) faster_rcnn c++版本的 caffe 封装（1）

10、 https://github.com/YihangLou/FasterRCNN-Encapsulation-Cplusplus (https://github.com/YihangLou/FasterRCNN-Encapsulation-Cplusplus)



11、http://blog.csdn.net/xyy19920105/article/details/50440957 (http://blog.csdn.net/xyy19920105/article/details/50440957) 将Faster RCNN的python demo改成C++ demo

12、http://m.blog.csdn.net/article/details?id=51036677 (http://m.blog.csdn.net/article/details?id=51036677) 目标检测 Fast RCNN算法详解

13、http://jingyan.baidu.com/article/eae07827f7f2d01fec5485f7.html (http://jingyan.baidu.com/article/eae07827f7f2d01fec5485f7.html) ubuntu 安装python

14、http://www.cnblogs.com/linkboy1980/p/5469994.html (http://www.cnblogs.com/linkboy1980/p/5469994.html) CAFFE+FASTERCNN安装记录

15、http://blog.csdn.net/ture_dream/article/details/52758422 (http://blog.csdn.net/ture_dream/article/details/52758422) 一张图理清CNN脉络 RCNN SPP FASTRCNN FASTERRCNN 到 MSCNN

补充资料：

https://github.com/rbgirshick/fast-rcnn (https://github.com/rbgirshick/fast-rcnn)

This code base is no longer maintained and exists as a historical artifact to supplement my ICCV 2015 paper. For more recent work that’s faster and more accurrate, please see Faster R-CNN (which also includes functionality for training Fast R-CNN).

Fast R-CNN: Fast Region-based Convolutional Networks for object detection

Created by Ross Girshick at Microsoft Research, Redmond.

Introduction

Fast R-CNN is a fast framework for object detection with deep ConvNets. Fast R-CNN

```
trains state-of-the-art models, like VGG16, 9x faster than traditional R-CNN and 3x faster
runs 200x faster than R-CNN and 10x faster than SPPnet at test-time,
has a significantly higher mAP on PASCAL VOC than both R-CNN and SPPnet,
and is written in Python and C++/Caffe.
```

Fast R-CNN was initially described in an arXiv tech report and later published at ICCV 2015.

License

Fast R-CNN is released under the MIT License (refer to the LICENSE file for details).

Citing Fast R-CNN

If you find Fast R-CNN useful in your research, please consider citing:

@inproceedings{girshickICCV15fastrcnn,

Author = {Ross Girshick},

Title = {Fast R-CNN},

Booktitle = {International Conference on Computer Vision ({ICCV})},

Year = {2015}

}

Contents

Requirements: software

```
Requirements for Caffe and pycaffe (see: Caffe installation instructions)

Note: Caffe must be built with support for Python layers!

# In your Makefile.config, make sure to have this line uncommented
WITH_PYTHON_LAYER := 1

You can download my Makefile.config for reference.
Python packages you might not have: cython, python-opencv, easydict
[optional] MATLAB (required for PASCAL VOC evaluation only)
```

Requirements: hardware

```
For training smaller networks (CaffeNet, VGG_CNN_M_1024) a good GPU (e.g., Titan, K20, K40
For training with VGG16, you'll need a K40 (~11G of memory)
```

Installation (sufficient for the demo)

```
Clone the Fast R-CNN repository

# Make sure to clone with --recursive
git clone --recursive https://github.com/rbgirshick/fast-rcnn.git

We'll call the directory that you cloned Fast R-CNN into FRCN_ROOT

Ignore notes 1 and 2 if you followed step 1 above.

Note 1: If you didn't clone Fast R-CNN with the --recursive flag, then you'll need to manu

git submodule update --init --recursive

Note 2: The caffe-fast-rcnn submodule needs to be on the fast-rcnn branch (or equivalent c

Build the Cython modules

cd $FRCN_ROOT/lib
make

Build Caffe and pycaffe

cd $FRCN_ROOT/caffe-fast-rcnn
# Now follow the Caffe installation instructions here:
#   http://caffe.berkeleyvision.org/installation.html

# If you're experienced with Caffe and have all of the requirements installed
# and your Makefile.config in place, then simply do:
make -j8 && make pycaffe

Download pre-computed Fast R-CNN detectors

cd $FRCN_ROOT
```

```
./data/scripts/fetch_fast_rcnn_models.sh

This will populate the $FRCN_ROOT/data folder with fast_rcnn_models. See data/README.md fo
```

Demo

After successfully completing basic installation, you'll be ready to run the demo.

Python

To run the demo

cd $FRCN_ROOT

./tools/demo.py

The demo performs detection using a VGG16 network trained for detection on PASCAL VOC 2007. The object proposals are pre-computed in order to reduce installation requirements.

Note: If the demo crashes Caffe because your GPU doesn't have enough memory, try running the demo with a small network, e.g., ./tools/demo.py –net caffenet or with –net vgg_cnn_m_1024. Or run in CPU mode ./tools/demo.py –cpu. Type ./tools/demo.py -h for usage.

MATLAB

There's also a basic MATLAB demo, though it's missing some minor bells and whistles compared to the Python version.

cd $FRCN_ROOT/matlab

matlab # wait for matlab to start…

At the matlab prompt, run the script:

fast_rcnn_demo

Fast R-CNN training is implemented in Python only, but test-time detection functionality also exists in MATLAB. See matlab/fast_rcnn_demo.m and matlab/fast_rcnn_im_detect.m for details.

Computing object proposals

The demo uses pre-computed selective search proposals computed with this code. If you'd like to compute proposals on your own images, there are many options. Here are some pointers; if you run into trouble using these resources please direct questions to the respective authors.

```
Selective Search: original matlab code, python wrapper
EdgeBoxes: matlab code
GOP and LPO: python code
MCG: matlab code
RIGOR: matlab code
```

Apologies if I've left your method off this list. Feel free to contact me and ask for it to be included.

Beyond the demo: installation for training and testing models

```
Download the training, validation, test data and VOCdevkit

wget http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCtrainval_06-Nov-2007.tar
wget http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCtest_06-Nov-2007.tar
wget http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCdevkit_08-Jun-2007.tar

Extract all of these tars into one directory named VOCdevkit

tar xvf VOCtrainval_06-Nov-2007.tar
tar xvf VOCtest_06-Nov-2007.tar
tar xvf VOCdevkit_08-Jun-2007.tar

It should have this basic structure

$VOCdevkit/                           # development kit
$VOCdevkit/VOCcode/                   # VOC utility code
$VOCdevkit/VOC2007                    # image sets, annotations, etc.
# ... and several other directories ...

Create symlinks for the PASCAL VOC dataset

cd $FRCN_ROOT/data
ln -s $VOCdevkit VOCdevkit2007

Using symlinks is a good idea because you will likely want to share the same PASCAL datase
[Optional] follow similar steps to get PASCAL VOC 2010 and 2012
Follow the next sections to download pre-computed object proposals and pre-trained ImageNe
```

Download pre-computed Selective Search object proposals

Pre-computed selective search boxes can also be downloaded for VOC2007 and VOC2012.

cd $FRCN_ROOT

./data/scripts/fetch_selective_search_data.sh

This will populate the $FRCN_ROOT/data folder with selective_selective_data.

Download pre-trained ImageNet models

Pre-trained ImageNet models can be downloaded for the three networks described in the paper: CaffeNet (model S), VGG_CNN_M_1024 (model M), and VGG16 (model L).

cd $FRCN_ROOT

./data/scripts/fetch_imagenet_models.sh

These models are all available in the Caffe Model Zoo, but are provided here for your convenience.

Usage

Train a Fast R-CNN detector. For example, train a VGG16 network on VOC 2007 trainval:

./tools/train_net.py –gpu 0 –solver models/VGG16/solver.prototxt \

–weights data/imagenet_models/VGG16.v2.caffemodel

If you see this error

EnvironmentError: MATLAB command ‘matlab’ not found. Please add ‘matlab’ to your PATH.

then you need to make sure the matlab binary is in your $PATH. MATLAB is currently required for PASCAL VOC evaluation.

Test a Fast R-CNN detector. For example, test the VGG 16 network on VOC 2007 test:

./tools/test_net.py –gpu 1 –def models/VGG16/test.prototxt \

–net output/default/voc_2007_trainval/vgg16_fast_rcnn_iter_40000.caffemodel

Test output is written underneath $FRCN_ROOT/output.

Compress a Fast R-CNN model using truncated SVD on the fully-connected layers:

./tools/compress_net.py –def models/VGG16/test.prototxt \

–def-svd models/VGG16/compressed/test.prototxt \

–net output/default/voc_2007_trainval/vgg16_fast_rcnn_iter_40000.caffemodel

Test the model you just compressed

./tools/test_net.py –gpu 0 –def models/VGG16/compressed/test.prototxt \

–net output/default/voc_2007_trainval/vgg16_fast_rcnn_iter_40000_svd_fc6_1024_fc7_256.caffemodel

Experiment scripts

Scripts to reproduce the experiments in the paper (up to stochastic variation) are provided in

$FRCN_ROOT/experiments/scripts. Log files for experiments are located in experiments/logs.

Note: Until recently (commit a566e39), the RNG seed for Caffe was not fixed during training. Now it’s fixed, unless

train_net.py is called with the –rand flag. Results generated before this commit will have some stochastic variation.

Extra downloads

```
Experiment logs
PASCAL VOC test set detections
    voc_2007_test_results_fast_rcnn_caffenet_trained_on_2007_trainval.tgz
    voc_2007_test_results_fast_rcnn_vgg16_trained_on_2007_trainval.tgz
    voc_2007_test_results_fast_rcnn_vgg_cnn_m_1024_trained_on_2007_trainval.tgz
    voc_2012_test_results_fast_rcnn_vgg16_trained_on_2007_trainvaltest_2012_trainval.tgz
    voc_2012_test_results_fast_rcnn_vgg16_trained_on_2012_trainval.tgz
Fast R-CNN VGG16 model trained on VOC07 train,val,test union with VOC12 train,val
```

本文地址：http://blog.csdn.net/forest_world (http://blog.csdn.net/forest_world)

查看原文>> (http://blog.csdn.net/forest_world/article/details/53036656)

2

看过本文的人也看了：

- 深度学习知识结构图 (http://lib.csdn.net/base/deeplearning/structure)
- Willow: A User-Programmable SSD (http://lib.csdn.net/article/deeplearning/48279)
- RCNN--对象检测的又一伟大跨越 2（包... (http://lib.csdn.net/article/deeplearning/61631)
- 使用faster rcnn训练自己的模型 (http://lib.csdn.net/article/deeplearning/57867)
- RCNN (Regions with CNN) 目标物检测... (http://lib.csdn.net/article/deeplearning/61653)
- YOLO源码详解（二）- 函数剖析 (http://lib.csdn.net/article/deeplearning/53854)

发表评论

输入评论内容

发表

0个评论