# deepLab训练

## 一 deeplab训练环境搭建

* 1. 参考

|  |
| --- |
| <https://lijiancheng0614.github.io/2018/03/13/2018_03_13_TensorFlow-DeepLab/> |

* 1. 数据集

准备[VOCtrainval\_11-May-2012.tar](http://geeekvr.com:18129/edit/work/deeplab/datasets/pascal_voc_seg/VOCtrainval_11-May-2012.tar)

* 1. 下载code

|  |
| --- |
| git clone <https://github.com/sycophant-stone/tf_base.git> |

精简模式

|  |
| --- |
| git clone --depth=1 https://github.com/sycophant-stone/tf\_base.git |

* 1. 把VOC放到research/deeplab/datasets中
  2. 运行[download\_and\_convert\_voc2012.sh](http://geeekvr.com:18129/edit/work/deeplab/datasets/download_and_convert_voc2012.sh)
  3. 使用slim的env环境设置

首先先解压缩(注意这里-o的用法)

|  |
| --- |
| root@df4112054141:/work/gi/tf\_base/research# **unzip -o protobuf.zip** |

然后运行工具脚本

|  |
| --- |
| #!/bin/bash  echo "export"  **export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim**  echo "protoc"  /home/julyedu\_433249/work/tf\_base/research/bin/protoc object\_detection/protos/\*.proto --python\_out=. |

* 1. 下载pretrain模型

|  |
| --- |
| # From deeplab/datasets/pascal\_voc\_seg/  mkdir init\_models  cd init\_models  wget http://download.tensorflow.org/models/deeplabv3\_pascal\_train\_aug\_2018\_01\_04.tar.gz  tar zxf ssd\_mobilenet\_v1\_coco\_11\_06\_2017.tar.gz |

下载dataset

|  |
| --- |
| http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCtrainval\_11-May-2012.tar |

* 1. train用的命令

|  |
| --- |
| python deeplab/train.py --logtostderr --training\_number\_of\_steps=30000 --train\_split="train" --model\_variant="xception\_65" --atrous\_rates=6 --atrous\_rates=12 --atrous\_rates=18 --output\_stride=16 --decoder\_output\_stride=4 --train\_crop\_size=513 --train\_crop\_size=513 --train\_batch\_size=1 --dataset="pascal\_voc\_seg" --tf\_initial\_checkpoint=deeplab/init\_models/deeplabv3\_pascal\_train\_aug/model.ckpt --train\_logdir=deeplab/trainlog --dataset\_dir=deeplab/datasets/pascal\_voc\_seg/tfrecord |
|  |

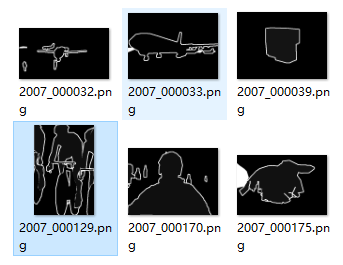
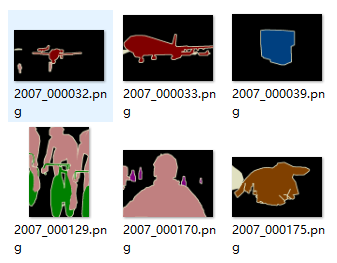
* 1. **win10上搭建**
     1. 搭建数据集

|  |
| --- |
| Gitbash中运行  ./download\_and\_convert\_voc2012.sh  利用其解压缩的方法. |

### 整理数据集

利用class来生成raw class.

|  |
| --- |
| cd D:\work\stuff\modules\misc\sprd\_camera\alg\july\tf\_base\research\deeplab\datasets  **python remove\_gt\_colormap.py** --original\_gt\_folder="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\VOCdevkit\\VOC2012\\SegmentationClass" --output\_dir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\VOCdevkit\\VOC2012\\SegmentationClassRaw" |



利用tf生成tfrecord

|  |
| --- |
| (tensorflow) D:\work\stuff\modules\misc\sprd\_camera\alg\july\tf\_base\research\deeplab\datasets>  **python** **build\_voc2012\_data.py** --image\_folder="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\VOCdevkit\\VOC2012\\JPEGImages" --semantic\_segmentation\_folder="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\VOCdevkit\\VOC2012\\SegmentationClassRaw" --list\_folder="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\VOCdevkit\\VOC2012\\ImageSets\\Segmentation" --image\_format=jpg --output\_dir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\tfrecord" |

### 可用的环境总结

|  |
| --- |
|  |

|  |
| --- |
| tf\_initial\_checkpoint = "D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\init\_models\\deeplabv3\_pascal\_train\_aug\\model.ckpt" |

**pyCharm使用调试golden的命令:**

|  |
| --- |
| python deeplab/train.py --logtostderr --training\_number\_of\_steps=30000 --train\_split="train" --model\_variant="xception\_65" --atrous\_rates=6 --atrous\_rates=12 --atrous\_rates=18 --output\_stride=16 --decoder\_output\_stride=4 --train\_crop\_size=513 --train\_crop\_size=513 --train\_batch\_size=1 --dataset="pascal\_voc\_seg" --tf\_initial\_checkpoint="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\init\_models\\deeplabv3\_pascal\_train\_aug\\model.ckpt" --train\_logdir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\output" --dataset\_dir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\tfrecord" |

### pyCharm中添加path

**PYTHONPATH**

|  |
| --- |
| D:\work\stuff\modules\misc\sprd\_camera\alg\july\tf\_base\research;D:\work\stuff\modules\misc\sprd\_camera\alg\july\tf\_base\research\slim |

## 二 deeplab的evaluation和test

2.1 eval方法

|  |
| --- |
| # From tensorflow/models/research/  python deeplab/eval.py \  --logtostderr \  --eval\_split="val" \  --model\_variant="xception\_65" \  --atrous\_rates=6 \  --atrous\_rates=12 \  --atrous\_rates=18 \  --output\_stride=16 \  --decoder\_output\_stride=4 \  --eval\_crop\_size=513 \  --eval\_crop\_size=513 \  --dataset="pascal\_voc\_seg" \  --checkpoint\_dir=${PATH\_TO\_CHECKPOINT} \  --eval\_logdir=${PATH\_TO\_EVAL\_DIR} \  --dataset\_dir=${PATH\_TO\_DATASET} |

Win10上命令

|  |
| --- |
| python deeplab/eval.py --logtostderr --eval\_split="val" --model\_variant="xception\_65" --atrous\_rates=6 --atrous\_rates=12 --atrous\_rates=18 --output\_stride=16 --decoder\_output\_stride=4 --eval\_crop\_size=513 --eval\_crop\_size=513 --dataset="pascal\_voc\_seg" --checkpoint\_dir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\output" --eval\_logdir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\eval\_output" --dataset\_dir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\tfrecord" |

参考结果:

|  |
| --- |
| INFO:tensorflow:Evaluation [1449/1449]  INFO:tensorflow:Finished evaluation at 2018-10-24-13:14:13  miou\_1.0[0.761658609] |

### 2.2 可视化方法

|  |
| --- |
| # From tensorflow/models/research/  python deeplab/vis.py \  --logtostderr \  --vis\_split="val" \  --model\_variant="xception\_65" \  --atrous\_rates=6 \  --atrous\_rates=12 \  --atrous\_rates=18 \  --output\_stride=16 \  --decoder\_output\_stride=4 \  --vis\_crop\_size=513 \  --vis\_crop\_size=513 \  --dataset="pascal\_voc\_seg" \  --checkpoint\_dir=${PATH\_TO\_CHECKPOINT} \  --vis\_logdir=${PATH\_TO\_VIS\_DIR} \  --dataset\_dir=${PATH\_TO\_DATASET} |

Win10上命令:

|  |
| --- |
| python deeplab/vis.py --logtostderr --vis\_split="val" --model\_variant="xception\_65" --atrous\_rates=6 --atrous\_rates=12 --atrous\_rates=18 --output\_stride=16 --decoder\_output\_stride=4 --vis\_crop\_size=513 --vis\_crop\_size=513 --dataset="pascal\_voc\_seg" --checkpoint\_dir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\output" --vis\_logdir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\vis\_log" --dataset\_dir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\tfrecord" |

## 三 可用的gpu环境

### 3.1 极客云

|  |
| --- |
| ssh -p 8065 [root@geeekvr.com](mailto:root@geeekvr.com)  ssh -L 16006:127.0.0.1:6006 -p 8065 [root@geeekvr.com](mailto:root@geeekvr.com) |

## 四 tensorboard使用方法

### 4.1 tensorboard映射

|  |
| --- |
| ssh -L 16006:127.0.0.1:6006 julyedu\_433249@101.132.121.198  tensorboard --logdir="/path/to/log-directory"  最后，在本地访问地址：http://127.0.0.1:16006/ |

### 4.2 win10上用法

第一步,打开anaconda下tensorflow环境的terminal.

第二步,运行

|  |
| --- |
| (tensorflow) C:\Users\raytine>***tensorboard --logdir="D:\\work\\stuff\\modules\\misc\\sprd\_camera\\alg\\july\\tf\_base\\research\\deeplab\\datasets\\pascal\_voc\_seg\\output\_new"*** |

第三步,打开网址

http://DESKTOP-58H7SAQ:6006

## 五 deeplab网络

### 5.1 deeplab的Loss

|  |
| --- |
| **def** add\_softmax\_cross\_entropy\_loss\_for\_each\_scale**(**scales\_to\_logits**,**  labels**,**  num\_classes**,**  ignore\_label**,**  loss\_weight**=**1.0**,**  upsample\_logits**=True,**  scope**=None):**  """Adds softmax cross entropy loss for logits of each scale.  **## 采用softmax做loss.**  **应该是每个像素做loss.**  Args:  scales\_to\_logits: A map from logits names for different scales to logits.  **## 预测值, [b h w num\_cls]**  The logits have shape [batch, logits\_height, logits\_width, num\_classes].  labels: Groundtruth labels with shape [batch, image\_height, image\_width, 1].  **## gt labels, [b ih iw 1]**  num\_classes: Integer, number of target classes.  ignore\_label: Integer, label to ignore.  loss\_weight: Float, loss weight.  upsample\_logits: Boolean, upsample logits or not.  scope: String, the scope for the loss.  Raises:  ValueError: Label or logits is None.  """  **if** labels **is** **None:**  **raise** ValueError**(**'No label for softmax cross entropy loss.'**)**  **for** scale**,** logits **in** six**.**iteritems**(**scales\_to\_logits**):**  loss\_scope **=** **None**  **if** scope**:**  loss\_scope **=** '%s\_%s' **%** **(**scope**,** scale**)**  **## 保证label和prediction是同尺寸的.**   1. **要么对logits做上采样.** 2. **要么对label做下采样.**   **if** upsample\_logits**:**  # Label is not downsampled, and instead we upsample logits.  logits **=** tf**.**image**.**resize\_bilinear**(**  logits**,**  preprocess\_utils**.**resolve\_shape**(**labels**,** 4**)[**1**:**3**],**  align\_corners**=True)**  scaled\_labels **=** labels  **else:**  # Label is downsampled to the same size as logits.  scaled\_labels **=** tf**.**image**.**resize\_nearest\_neighbor**(**  labels**,**  preprocess\_utils**.**resolve\_shape**(**logits**,** 4**)[**1**:**3**],**  align\_corners**=True)**  **## 采用softmax**   1. **样本不均衡时,可以设置不同label的weights. 让样本少的class,获得大一些的权重值.** 2. **Softmax一方面是labels(经过独热编码后),另一方面是预测logits.**   scaled\_labels **=** tf**.**reshape**(**scaled\_labels**,** shape**=[-**1**])**  not\_ignore\_mask **=** tf**.**to\_float**(**tf**.**not\_equal**(**scaled\_labels**,**  ignore\_label**))** **\*** loss\_weight  one\_hot\_labels **=** slim**.**one\_hot\_encoding**(**  scaled\_labels**,** num\_classes**,** on\_value**=**1.0**,** off\_value**=**0.0**)**  tf**.**losses**.**softmax\_cross\_entropy**(**  one\_hot\_labels**,**  tf**.**reshape**(**logits**,** shape**=[-**1**,** num\_classes**]),**  weights**=**not\_ignore\_mask**,**  scope**=**loss\_scope**)** |

5.2 多尺度预测

|  |
| --- |
| **def** multi\_scale\_logits**(**images**,**  model\_options**,**  image\_pyramid**,**  weight\_decay**=**0.0001**,**  is\_training**=False,**  fine\_tune\_batch\_norm**=False):**  """Gets the logits for multi-scale inputs.  **## 拿到多尺度输入的预测值.**   1. **由于max pooling,这些预测值们是”降采样的”** 2. **这种降采样对train和eval都是存在的.**   The returned logits are all downsampled (due to max-pooling layers)  for both training and evaluation.  Args:  **images**: A tensor of size **[batch, height, width, channels]**.  model\_options: A ModelOptions instance to configure models.  image\_pyramid: Input image scales for multi-scale feature extraction.  weight\_decay: The weight decay for model variables.  is\_training: Is training or not.  fine\_tune\_batch\_norm: Fine-tune the batch norm parameters or not.  Returns:  **outputs\_to\_scales\_to\_logits**: A map of maps from output\_type (e.g., semantic prediction) to a dictionary of multi-scale logits names to logits. For each output\_type, the dictionary has keys which correspond to the scales and values which correspond to the logits. For example, if `scales` equals [1.0, 1.5], then the keys would include 'merged\_logits', 'logits\_1.00' and 'logits\_1.50'.  **## 输出logits的scales.**   1. **一组maps.** 2. **从输出类型(语义预测) 到和logits单元相关的多尺度预测值组成的字典的映射(map).** 3. **对于每一个输出类型, 字典都含有一组keys值.** 4. **这组keys值对应于scales和values** 5. **这些scales和values描述对应的预测值(logits).** 6. **例如, scales为[1.0, 1.5],其含义是,字典的keys是 merged\_logits, logits\_1.00和logits\_1.50**   Raises:  ValueError: If model\_options doesn't specify crop\_size and its  add\_image\_level\_feature = True, since add\_image\_level\_feature requires  crop\_size information.  """ |

|  |
| --- |
| # Setup default values.  **if** **not** image\_pyramid**:**  image\_pyramid **=** **[**1.0**]**  crop\_height **=** **(**  model\_options**.**crop\_size**[**0**]**  **if** model\_options**.**crop\_size **else** tf**.**shape**(**images**)[**1**])**  crop\_width **=** **(**  model\_options**.**crop\_size**[**1**]**  **if** model\_options**.**crop\_size **else** tf**.**shape**(**images**)[**2**])**  # Compute the height, width for the output logits.  logits\_output\_stride **=** **(**  model\_options**.**decoder\_output\_stride **or** model\_options**.**output\_stride**)**  logits\_height **=** scale\_dimension**(**  crop\_height**,**  max**(**1.0**,** max**(**image\_pyramid**))** **/** logits\_output\_stride**)**  logits\_width **=** scale\_dimension**(**  crop\_width**,**  max**(**1.0**,** max**(**image\_pyramid**))** **/** logits\_output\_stride**)**  # Compute the logits for each scale in the image pyramid.  outputs\_to\_scales\_to\_logits **=** **{**  k**:** **{}**  **for** k **in** model\_options**.**outputs\_to\_num\_classes  **} ## 计算多尺度中每个尺度对应的scales值(keys).**  **for** image\_scale **in** image\_pyramid**:**  **if** image\_scale **!=** 1.0**:**  scaled\_height **=** scale\_dimension**(**crop\_height**,** image\_scale**)**  scaled\_width **=** scale\_dimension**(**crop\_width**,** image\_scale**)**  scaled\_crop\_size **=** **[**scaled\_height**,** scaled\_width**]**  scaled\_images **=** tf**.**image**.**resize\_bilinear**(**  images**,** scaled\_crop\_size**,** align\_corners**=True)**  **if** model\_options**.**crop\_size**:**  scaled\_images**.**set\_shape**([None,** scaled\_height**,** scaled\_width**,** 3**])**  **else:**  scaled\_crop\_size **=** model\_options**.**crop\_size  scaled\_images **=** images  updated\_options **=** model\_options**.**\_replace**(**crop\_size**=**scaled\_crop\_size**)**  outputs\_to\_logits **=** **\_get\_logits( ## 拿到predictions( logits tensor)**  scaled\_images**,**  updated\_options**,**  weight\_decay**=**weight\_decay**,**  reuse**=**tf**.**AUTO\_REUSE**,**  is\_training**=**is\_training**,**  fine\_tune\_batch\_norm**=**fine\_tune\_batch\_norm**)**  # Resize the logits to have the same dimension before merging.  **for** output **in** sorted**(**outputs\_to\_logits**): ## 多尺度要有相同的size.**  outputs\_to\_logits**[**output**]** **=** tf**.**image**.**resize\_bilinear**(**  outputs\_to\_logits**[**output**],** **[**logits\_height**,** logits\_width**],**  align\_corners**=True)**  # Return when only one input scale.  **if** len**(**image\_pyramid**)** **==** 1**:**  **for** output **in** sorted**(**model\_options**.**outputs\_to\_num\_classes**):**  outputs\_to\_scales\_to\_logits**[**output**][**  MERGED\_LOGITS\_SCOPE**]** **=** outputs\_to\_logits**[**output**]**  **return** outputs\_to\_scales\_to\_logits  # Save logits to the output map.  **for** output **in** sorted**(**model\_options**.**outputs\_to\_num\_classes**):**  outputs\_to\_scales\_to\_logits**[**output**][**  'logits\_%.2f' **%** image\_scale**]** **=** outputs\_to\_logits**[**output**]**  # Merge the logits from all the multi-scale inputs.  **for** output **in** sorted**(**model\_options**.**outputs\_to\_num\_classes**):## 合并所有的多尺度logits.**  # Concatenate the multi-scale logits for each output type.  all\_logits **=** **[**  tf**.**expand\_dims**(**logits**,** axis**=**4**) ## 在dim4上扩充1维度(为了concat).**  **for** logits **in** outputs\_to\_scales\_to\_logits**[**output**].**values**()**  **]**  all\_logits **=** tf**.**concat**(**all\_logits**,** 4**)##在dim4上concat.**  merge\_fn **=** **(**  tf**.**reduce\_max  **if** model\_options**.**merge\_method **==** 'max' **else** tf**.**reduce\_mean**)**  outputs\_to\_scales\_to\_logits**[**output**][**MERGED\_LOGITS\_SCOPE**]** **=** **merge\_fn(**  all\_logits**,** axis**=**4**) ## 消除dim4, 采用reduce\_max或reduce\_mean的方法.**  **return** outputs\_to\_scales\_to\_logits |

#### 何为\_get\_logits?

## 六 答疑

### 6.1 VOC图片原尺寸,以及crop后的尺寸

原尺寸:

Crop后尺寸可调,目前是513,513

### 6.2 数据不均衡现象

修改utils下train\_utils.py

     由于数据集不平衡，对loss的权重系数进行修改。经过计算，像素比 px(0):px(1)=15:1, 所以取 label0\_weight = 1, label1\_weight = 15.

# 训练自己的数据集，针对数据不平衡，此处进行修改

|  |
| --- |
| ignore\_weight = 0  label0\_weight = 1 # 背景的权重系数  label1\_weight = 15 # \*\* 的权重系数      not\_ignore\_mask = tf.to\_float(tf.equal(scaled\_labels, 0)) \* label0\_weight + \  tf.to\_float(tf.equal(scaled\_labels, 1)) \* label1\_weight + \  tf.to\_float(tf.equal(scaled\_labels, ignore\_label)) \* ignore\_weight    tf.losses.softmax\_cross\_entropy(  one\_hot\_labels,  tf.reshape(logits, shape=[-1, num\_classes]),  weights=not\_ignore\_mask,  scope=loss\_scope)      同时，修改 exclude\_list。  exclude\_list = ['global\_step', 'logits'] # 训练自己的数据集时，此处进行修改 |

### 6.3 怎么训练自己的网络以及数据集.

目的是做一个二分类, 利用pretrained模型, fine tune最后一层logits.

|  |
| --- |
| # Variables that will not be restored.  exclude\_list = ['global\_step','logits'] **## 不加载logits层(logits层不采用pretrained参数).**  #exclude\_list = ['global\_step']  if not initialize\_last\_layer:  exclude\_list.extend(last\_layers) |

然后在自己的数据集上fine tune.

需要制作自己数据集的tfrecord.以及读取数据集的配置.

|  |
| --- |
| \_SCREW = DatasetDescriptor(  splits\_to\_sizes={  'train': 119, # num of samples in images/training  #'train\_aug': 10582,  #'trainval': 2913,  #'val': 3000,  },  **num\_classes=3, #background、ignore\_label、ignore\_label，即label数+2(含背景).**  ignore\_label=255,  ) |

### 6.4 mobileNetV2和deeplabV3的aspp和decode组合的训练

由于默认的mobileNetv2和deeplabv3没有aspp和decode部分.需要加上这部分做二分类训练.

1. 构造网络mobilenetV2+aspp+decode
2. 新网络在VOC2012上train
3. 然后在COCO上做train.
4. 经过前几步,就得到一个pretrained的模型,然后在固定logits之前的权重, 对logits在自己数据集上做fine tune.

## 七 实验

### 7.1 数据读取实验

描述: 从VOC格式数据,整理成tfrecord模式,并读入到tf的queue中.

### 7.2 数据增强实验

描述: 采用多种方式对train数据做数据增强.

### 7.3 xception网络搭建实验.

### 7.4 aspp网络搭建

### 7.5 decoder网络搭建

### 7.6 multi-scale网络实现

### 7.7 损失函数及评价函数实验

2018-10-12: 损失函数思路摸排好了. 评价函数还没找到.

### 7.8 summary实验

## 八 附录1 win10上搭建tenserflow-gpu版本

F1.1 安装anaconda.

F1.2 安装tensorflow-gpu

python install tensorflow-gpu

F1.3 安装对应的CUDA比如9.0

F1.4 然后尝试import tensorflow

会有提示,缺少一个` Could not find cudnn64\_7.dll`,下载对应的cudnn

F1.4 下载cudnn

<https://developer.nvidia.com/rdp/cudnn-download>

解压之后,把对应的cudnn64\_7.dll放到CUDA的bin下,应为CUDA的bin你也放到path路径上了.

## 九 附录二

Tf.TFRecordReader

slim.dataset\_data\_provider

tf.train.MomentumOptimizer

tf.image.resize\_images