

# Cross-domain Detection via Graph-induced Prototype Alignment 代码调试问题记录

[CVPR 2020](#)

[CODE](#)

## 1. 环境问题

按照要求的版本配置的

- conda 环境
  - 新建一个虚拟环境

```
1 | conda create -n GPA python=3.5
```

- 激活环境

```
1 | source activate GPA
```

- cuda安装
  - 选择对应的版本下载 [官网地址](#)
  - 安装，其中第一个驱动选择N，接下来一路Y

```
1 | sh cudaXXX.run
```

- 修改 ~/.bashrc，配置环境变量

```
1 | export CUDA_HOME=/usr/local/cuda
2 | export
   LD_LIBRARY_PATH=/usr/local/cuda/lib64:"$LD_LIBRARY_PATH:/usr/local/c
   uda/lib64:/usr/local/cuda/extras/CUPTI/lib64"
3 | export PATH=/usr/local/cuda/bin:$PATH
```

- cuDNN安装
  - 选择对应的版本下载 [官网地址](#)
  - 解压之后，将cuda/include/文件复制到usr/local/cuda/include文件夹，将cuda/lib64/下所有文件复制到usr/local/cuda/lib64文件夹中
  - 检查，nvcc -V
- PyTorch 0.4.0 安装

```
1 | pip install torch==0.4.0
```

- 其他包的安装，pip install即可，但是要注意scipy要安装1.2.0版本的

## 2. 编译问题

编译之后，运行的时候遇到了下面这个报错

```
1 | cudaCheckError() failed : no kernel image is available for execution on the device
```

我的实验环境：

pytorch0.4.0 cuda9.0 cudnn7.1.2 Tesla V100

原因[参考这里](#)

根据[链接](#)，将lib/make.sh的CUDA\_ARCH更新

```
1 | CUDA_ARCH="-gencode arch=compute_30,code=sm_30 \  
2 |             -gencode arch=compute_35,code=sm_35 \  
3 |             -gencode arch=compute_50,code=sm_50 \  
4 |             -gencode arch=compute_52,code=sm_52 \  
5 |             -gencode arch=compute_60,code=sm_60 \  
6 |             -gencode arch=compute_61,code=sm_61 \  
7 |             -gencode arch=compute_70,code=sm_70 \  
8 |             -gencode arch=compute_70,code=compute_70"
```

再次编译，即可

## 3. 使用自己的数据集

以train\_baseline.py为例

- 在train\_baseline.py中添加elif的数据集
- 在lib/dataset下面模仿sim10k.py增加自己数据集的文件
- 在lib/dataset/factory.py下增加数据集信息