

2019 未来杯高校AI挑战赛 区域赛作品

- 战队编号：10
- 战队名称：超越
- 战队成员：

概述

以Faster R-CNN为基础进行改进，包括数据预处理，运行，测试等。使用的基础网络是Resnet101。其中数据集进行了扩充，提升了检测效果。另外调整了模型的参数（如 `learning rate`、`weight decay` 等），提高了检测的精度和模型泛化能力。

系统要求

硬件环境要求

- CPU：4核以上
- GPU：TITAN Xp 尽量在12G以上
- 内存：8G以上
- 硬盘：50G以上
- 其他：无

软件环境要求

- 操作系统：Ubuntu16.04
- anaconda 4.6.9
- pytorch0.4.0
- cython
- cffi
- opencv-python
- scipy
- msgpack
- easydict
- matplotlib
- pyyaml
- tensorboardX
- cuda 9.0
- cudnn 7.5
- tensorflow
- tqdm

需要进行编译：

```
# 到code主目录下，然后运行以下程序进行编译
conda create -n pytorch python=3.6

# pytorch必须是0.4.0，可以从官网下载，然后离线安装
conda install pytorch torchvision cudatoolkit=9.0 -c pytorch=0.4.0

# 以上大部分需要软件都可以通过该命令执行
pip install -r requirements.txt
cd lib

# 在此处进行编译，如果遇到问题，修改CUDA_ARCH对应的参数即可
sh make.sh
```

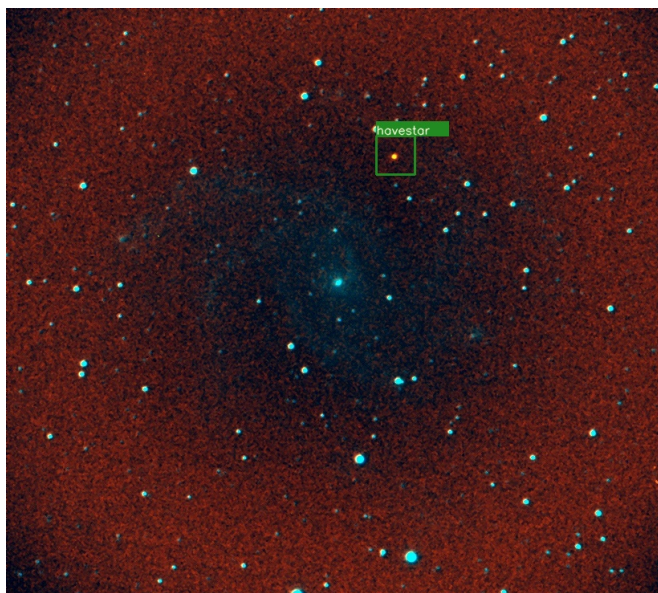
数据集


没有使用官方提供的数据集以外的数据。

数据预处理

方法概述

- 将提供的图片分为两类，分别为存在疑似星体与非星体的图片，将疑似星体类别中以"_a"、"_b"、"_c"为后缀的三张图片作为三个通道（依次对应R、G、B分量），合成一张图片，共计1049张，如下图所示。



- 将上述处理的图片，以疑似星体目标坐标为中心切割100*100的区域，然后使用双三次差值方法放大2倍得到200*200的图片，共计1049张，如下图所示。
-  将上述两类图片合并得到最终数据集，共2098张图片，全部用于训练。
- 数据集标注文件使用pascal_voc格式，其中xml标注文件中ground truth以目标坐标为中心，大小设为48*48（经测试尺寸为48*48时检测效果最好）。

操作步骤

- 使用 `moveFile()`，`mergeImg()` 函数将提供的数据集移到一个文件夹下且生成三通道的图片。

2. 使用 `makeVOCdevkit2007()` , `make_voc()` , `generateVOC()` 函数生成2098张图片的训练数据, 格式为 `pascal_voc`。
3. 以上函数代码均已集成至训练脚本 `train_net.py` 中。

模型

训练后的模型存储地址: 可以用 `--model` 参数进行指定, 默认位置为 `models`

模型文件大小: 377.5M

使用我们提供的训练好的权重: `faster_rcnn_1_10_499.pth`, 其位置在 `models/res101/pascal_voc/` 下, 运行 `demo.py` 时候, 使用 `--model models` 即可

训练

训练方法概述

- 基础网络选择 `resnet101`
- RPN部分anchor设置: `base_size=16, ratios=[0.5, 1, 2], scales=2*np.arange(3, 6)` (当前网络经测试取得的最好效果)
- 学习率设置为 `learning_rate=0.001`,
`learning_rate_decay_ratio=0.2, step_to_do_learning_rate_decay=5epoch`
- 训练优化方法为随机梯度下降 `stochastic gradient descent`
- 区域特征聚集方式使用 `roi_align`
- 训练动量 `train.momentum=0.9`
- 将所有图片按照比例缩放至600*1000的范围内
- `batch_size=3`
- `weight_decay=0.0001`

训练操作步骤

```
python trainval_net.py \  
    --net res101 \  
    --bs 3 \  
    --cuda \  
    --training-dataset=./af2019-cv-training-20190312 \  
    --model=model_dir
```

参数解释:

`--net` 选择基础网络, 我们使用的是 `res101`

`--bs` batch size, 每次放进网络三张图

`--cuda` 使用GPU

`--training-dataset` 训练集位置, 只需要跟官方提供的训练集一致即可, 即该文件夹下有許多子文件夹, 子文件夹中放置所有的数据图片, 该文件夹下需要有 `list.csv` 文件

`--model` 生成模型的位置, 默认地址是 `models`

训练结果保存与获取

训练结果直接保存在 `--model` 指定的文件夹中，该文件夹结构如下：

```
- models
  - res101
    - pascal_voc
      - faster_rcnn_1_1_499.pth
      - faster_rcnn_1_1_698.pth
      - faster_rcnn_1_2_499.pth
      - faster_rcnn_1_2_698.pth
      - faster_rcnn_1_3_499.pth
      - ...
      - faster_rcnn_1_10_499.pth
      - faster_rcnn_1_10_698.pth
```

训练权重格式为 `faster_rcnn_x_y_z.pth`，其中，`x`代表checksession，`y`代表checkepoch，`z`代表checkpoint。

指定文件夹存储了许多checkpoint，一般选择 `faster_rcnn_1_10_499.pth`，经测试与其他模型相比效果更好。

测试

方法概述

将 `--image_dir` 对应的文件夹中的图片进行测试（即testB所在路径），使用 `demo.py` 文件进行测试，最终目的是得到 `submit.csv` 文件。

- `RPN_NMS_THRESH=0.7`，`RPN_POST_NMS_TOP_N=500`，`RPN_PRE_NMS_TOP_N=10000`
- 区域特征聚集方式使用 `roi_align`

测试最优权重的选择参考TensorboardX展示的Loss，Train/Accuracy及权重直方图三者综合确定。最优权重的Loss变化缓慢，Accuracy值较高，权重稳定。同时过拟合因素也考虑在内，未选择最后一个权重，所以最终选择的权重文件为 `faster_rcnn_1_10_499.pth`。

操作步骤

如果当前有 `submit.csv` 文件，先删除上一次得到的 `submit.csv`

```
python demo.py --net res101 --cuda \
    --checksession 1 \
    --checkepoch 10 \
    --checkpoint 499 \
    --prediction-file=./submit.csv \
    --model=models \
    --test-dataset=./af2019-cv-testB-20190408/ \
    --image_dir ./images/
```

参数解释：

`--net` 选择基础网络为res101

`--cuda` 使用GPU

--checksession , --checkepoch , --checkpoint 三个参数分别对应上述checkpoint文件的命名，如
faster_rcnn_1_10_499.pth 对应 --checksession 1 --checkepoch 10 --checkpoint 499

--prediction-file 测试结果的csv文件及其路径

--test-dataset 指定测试集图片，与训练集类似，都会进行自动化处理，只需要解压testB，并且指向该目录即可。

--image_dir 指定处理后图片的输出位置