

# FH 模型调优训练工具

---

FH 模型调优训练工具旨在帮助开发者更快更好地完成人车，人形，人脸等物体检测模型的训练、优化、评估及部署等流程

## 目录

---

- [安装说明](#)
- [更新](#)
- [快速使用](#)

## 安装说明

---

### 环境需求

- Ubuntu 18.04 LTS
- Python 3.6.8
- CUDA 10.0

### 其他依赖安装

```
pip install -r requirements.txt
```

## 更新

---

- [2021.01.21]
  1. 增加图片格式支持 至 png jpg jpeg tiff tif bmp 格式
  2. 支持随机打乱或顺序生成tfrecords，为便于检查，会产生对应的图片次序文档
  3. 生成tfrecords过程中，支持打印正在处理的图片文件名,以方便定位
  4. 检查数据从展示图片模式改为保存图片模式
  5. 测试不需要将人和车分开，支持同时检测
  6. 修正了图片匹配方式
  7. 增加了生成芯片寄存器配置.nbg(nn binary graph)文件
- [2021.02.03]
  1. **添加人形检测训练和测试模块**
  2. 修正了训练测试结束后的TFError的报错
  3. 训练学习率策略不再要求固定长度，但要全部为浮点格式
- [2021.04.16]
  1. 更新了人车模型，在人车模型中增加了可配的loss比例，目前推荐1.0：1.0

2. 训练过程中，可以打印更详细的两类分类loss
3. 训练中会在当前目录下生成fullhan\_nn\_train.log，内容为loss信息，供后续分析使用

- [2021.04.26]

1. 加入了人脸检测，数据集图片数据只支持jpeg或jpg格式
2. 人脸检测一般会产生多个tfrecord，生成和使用时只需指定tfrecord所在的文件夹即可
3. **目前人脸检测只支持1个GPU训练**
4. 人车数据集画框验证改文字区分为红蓝颜色区分，更加直观

- [2021.05.28]

1. 修复人脸检测中的测试错误

- [2021.06.01]

1. 新的人车网络

## 快速使用

---

### 0. 训练配置

训练不同网络需要将对应的config\_xxx.py 重命名为 config.py，并在config.py里面修改数据路径和GPU相关配置

### 1. 准备数据

将图片和标签转化为深度学习可识别的数据格式

```
# 修改config.py里图片路径 config.image_file_dir 和标签路径 config.ann_file_dir, 打包生成后续训练所需要的数据格式
# 要求图片路径和标签存放在一级目录下，不支持将没有标签文件的图片生成数据集
# 图片路径 config.image_file_dir 内的图片支持 png jpg jpeg tiff tif bmp格式(人脸检测略微不同)
# 标签路径 config.ann_file_dir 为图片所对应的的标签txt所存放的路径，标签文件名和图片文件名必须对应一致。
# 标签内容为 “类别 x1 y1 x2 y2” 坐标为归一化值，类别0为人，车为1，人脸为2，空格为分隔符
# 保存路径为 config.tfrecord_path
# 是否随机打乱数据集 config.shuffle_enable, 输出一个txt:
# config.imgname_index_in_tfrecords, 内容为数据集内图片列表的顺序
# 图片和标签可参考ann_file文件夹和img_file文件夹
```

```
from config import config

if config.mode == 0: # detect car&person or person
    import sys
    sys.path.append('./tool')
if config.mode == 1: # detect car&person or person
    import sys
```

```

sys.path.append('./tool')
if config.mode == 2: # detect face
    import sys
    sys.path.append('./Face')# make sure 'Face' dir is in place

import fullhan_nn
fullhan_nn.tfrecords(config)
print('create tfrecords finished!!!')

```

参见 create\_tfrecords\_example.py

程序运行时，屏幕会持续输出打印正在处理的图片个数

如有图片与标签没有对应上，会提示图片路径

如图：

1.jpg 不存在对应的标签文件，故剔除，其余图片可以读入，并且人车均能读入

```

*** collecting all img and ann files, please wait 1 to 2 mins ***
4 pics in your dir
start generating tfrecords
*****
./img/2.jpg
num-pic: 1  num-person: 1  num-car: 0
*****
./img/4.jpg
num-pic: 2  num-person: 4  num-car: 0
*****
./img/3.jpg
num-pic: 3  num-person: 5  num-car: 1
*****
Found no ann file of img: ./img/1.jpg
*****
3 pics in tfrecords
./shuffled_imgname_.txt done!!!

```

程序结束后，会得到一个图片列表和一个打包数据集

图片列表如图所示：

```

|./img/2.jpg
|./img/4.jpg
|./img/3.jpg

```

## 2. 检查数据是否生成正确

保存 config.save\_pic\_num 个数的图片及检测框以检查数据是否正确，保存路径为 config.save\_pic\_path

```

# 保持上步保存路径为 config.tfrecord_path
from config import config

```

```

if config.mode == 0: # detect car&person or person
    import sys
    sys.path.append('./tool')
if config.mode == 1: # detect car&person or person
    import sys
    sys.path.append('./tool')
if config.mode == 2: # detect face
    import sys
    sys.path.append('./Face')# make sure 'Face' dir is in place

import fullhan_nn
fullhan_nn.save_img_from_tfreords(config)

```

参见：check\_data\_created\_correctness\_example.py

### 3. 开始训练

- 建议GPU Memory > 8G
- 确保上阶段生成的数据路径正确。
- 模型输出地址 config.checkpoint\_dir 可自定义
- **config.ckpt 为初始化的参数，可以在各自文件夹内找到。**（ must have ）

```

# 可使用默认的训练参数
# 使用推荐的初始化模型config.ckpt
from config import config

if config.mode == 0: # detect car and person
    import sys
    sys.path.append('./Car_Person')
if config.mode == 1: # detect person
    import sys
    sys.path.append('./Person')
if config.mode == 2: # detect face
    import sys
    sys.path.append('./Face')# make sure 'Face' dir is in place

import fullhan_nn
fullhan_nn.quant_train(config)

```

参见：finetune\_example.py

训练过程中,如果报错信息如下，说明参数设置有误，建议使用提供的默认参数：

```
***TFError***
```

训练过程中打印信息：

```

（时间）
step=( ), learn_rate=( ), loss_total=( ), loss_class_person=( ), loss_class_car=(
)

```

```

step=10, learn_rate=0.000000, loss_total=16.58, loss_class_person=4.07, loss_class_car=5.79
4.400858640670776
step=20, learn_rate=0.000000, loss_total=28.69, loss_class_person=6.62, loss_class_car=11.32
4.41510009765625
step=30, learn_rate=0.000000, loss_total=4.67, loss_class_person=2.12, loss_class_car=1.16
4.494095802307129
step=40, learn_rate=0.000000, loss_total=58.92, loss_class_person=5.22, loss_class_car=51.13
4.488421201705933
step=50, learn_rate=0.000000, loss_total=21.07, loss_class_person=5.42, loss_class_car=11.34
4.436811923980713

```

## 4. 测试

- 将训练完成的模型 通过修改config.py 中的 config.ckpt\_test 载入文件
- 输出测试结果config.output\_txt\_file

```

# config.ckpt_test 测试的模型文件
# config.img_path_for_test 支持两种模式：
# 1. 输入图片所在文件夹路径，自动解析文件夹下所有图片。
# 2. 输入待测试图片的路径列表，txt形式
# P.S. 图片格式支持 png jpg jpeg tiff tif bmp格式

from config import config

if config.mode == 0: # detect car and person
    import sys
    sys.path.append('./Car_Person')
if config.mode == 1: # detect person
    import sys
    sys.path.append('./Person')
if config.mode == 2: #detect face
    import sys
    sys.path.append('./Face')# make sure 'Face' dir is in place

import fullhan_nn
fullhan_nn.test(config)

```

测试过程中，终端打印所使用的的模型文件  
测试文件夹及里面图片的个数，以及处理的进度，如图所示

```

load succeeded ./initial_fullhan_checkpoint_20210121.bin
test_ckpt restore succeeded
Directory img_path_list /home/jinyc399/Desktop/share/liuxy/MsC0C02017/val2017_s
c, with 5000 images
19%|██████████          | 935/5000 [00:26<01:42, 39.77it/s]

```

输出txt格式

内容为 “图片 框个数” “类别 置信度 x y w h”

如图所示：

```
1 000000528524.jpg 185
2 0,0.122155,440,229,37,109
3 0,0.122155,434,445,41,79
4 0,0.122015,102,246,142,106
5 0,0.121861,131,77,41,20
6 0,0.119989,287,341,183,274
7 0,0.119989,50,44,419,291
8 0,0.118138,172,124,24,21
9 0,0.117849,128,52,192,86
```

## 5. 生成nbg文件

- 选择需要转化的模型地址写入 config.py 中config.ckpt\_to\_nbg
- 会在当前目录下生成nbg文件 c2det.nbg

```
from config import config

if config.mode == 0: # detect car and person
    import sys
    sys.path.append('./Car_Person')
if config.mode == 1: # detect person
    import sys
    sys.path.append('./Person')
if config.mode == 2: # detect face
    import sys
    sys.path.append('./Face')# make sure 'Face' dir is in place

import fullhan_nn
fullhan_nn.trans_ckpt_to_fullhan_nnip(config)
```

如图所示：

```
load ./initial_fullhan_checkpoint_20210121.bin succeeded
restoring variables...
[#####] 100% restore succeeded
analyzing...
[#####] 100% analyze completed

*** transform params in net ***
[#####] 100% transform completed

*** generate hardware .nbg ***
serializing completed
saving c2det.nbg ...
done!!!
```

## 错误解决

## 环境错误

### **libcublas.so 缺失**

cuda未安装正确。可尝试用conda，安装cuda。 `conda install cudatoolkit==10.0`

务必先修改config再进入程序运行，否则修改无效