

- 战队：业余玩家
- 队员：Derrors（队长）、心态巨烂、zzzlx
- 最终 B 榜成绩：0.90620 排名：6
- 运行环境：Ubuntu 16.0 + 1080Ti GPU Cudnn 9.0
- 使用的模型及框架：Darknet YOLOv3
- 开源代码：<https://github.com/pjreddie/darknet>
- 解题思路：

本模型基于 Darknet 深度学习框架的 YOLOv3 模型修改，根据比赛任务，修改模型源代码从而实现图片中零售商品的检测；

1. 数据集制作：

根据模型特点，将本次比赛的数据处理成模型接受的输入数据形式（VOC 数据集形式）；

2. 配置及源码修改：

根据任务需求，更改 cfg 目录下 voc.data、yolov3-voc.cfg 模型配置文件；

修改 data 目录下 voc.names 文件，设置类别名；

修改 examples 目录下 detector.c 代码，修改模型训练、测试过程代码并添加参数保存及结果输出；

3. 参数设置：yolov3-voc.cfg

训练过程： batch=64 subdivisions=8 learning_rate=0.03 random=1

微调过程： batch=32 subdivisions=8 learning_rate=0.0001

random=0

测试过程： batch=1 subdivisions=1

4. 目录及代码说明:

- 1) `./examples` 存放模型源代码
- 2) `./include` 存放模型代码头文件
- 3) `./data/voc.names` 记录类别名为文件
- 4) `./cfg` 存放模型参数及配置文件
- 5) `./backup/yolov3-voc_61161.weights` 最终结果的模型参数
- 6) `Makefile` 运行以编译模型
- 7) 生成当前目录下的文件名.BAT 生成所有图像名并写入 txt 文件的脚本
- 8) `read_data.py` 读取图像名称并生成训练集目录 txt
- 9) `generate_labels.py` 读取 `position.csv` 文件并生成每个目标的类别、box 坐标数据;
- 10) `trans_output.py` 将模型测试过程中生成的预测 `result.txt` 文件转换为 `output.csv` 文件
- 11) `output_61161.csv` 模型对测试集的最终结果

5. 注:

运行模型进行测试时,需预先运行 `MakeFile` 文件进行编译并修改配置文件中的图片、标签等的路径信息;