



# Object Detection

从零开始的七天AI之旅。

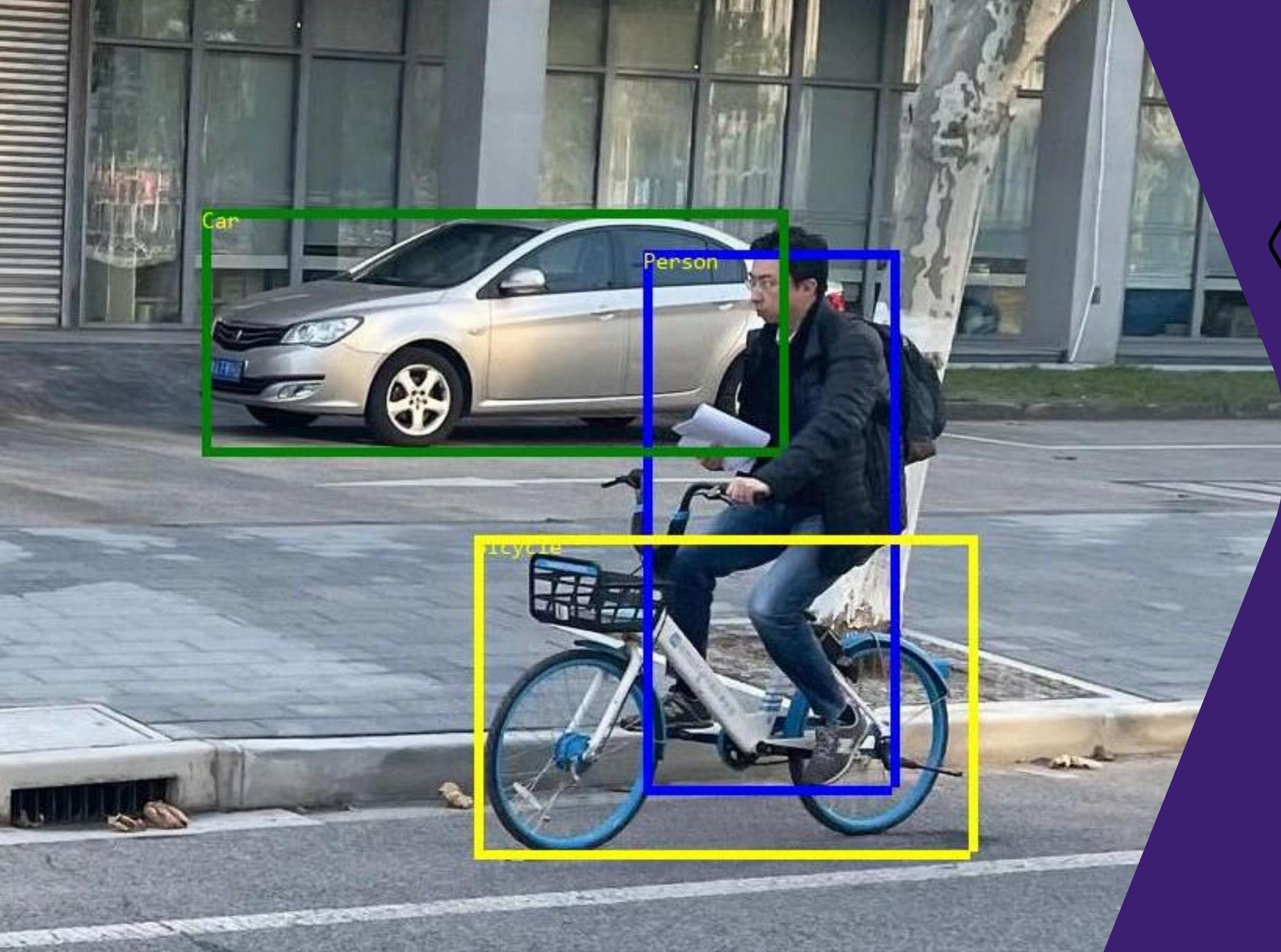
Group 6: 成笑行 陈奕澄 李博宇 庞皓天 王钧涛 张弛 朱泽凯



# 效果

自我感觉还不错。

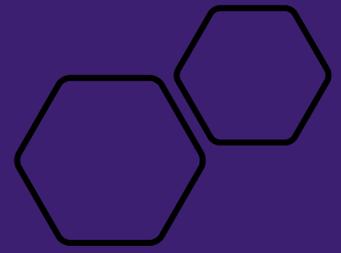




Car

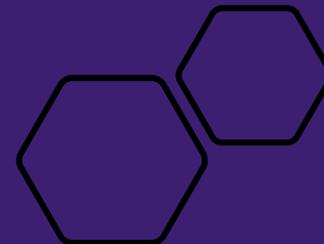
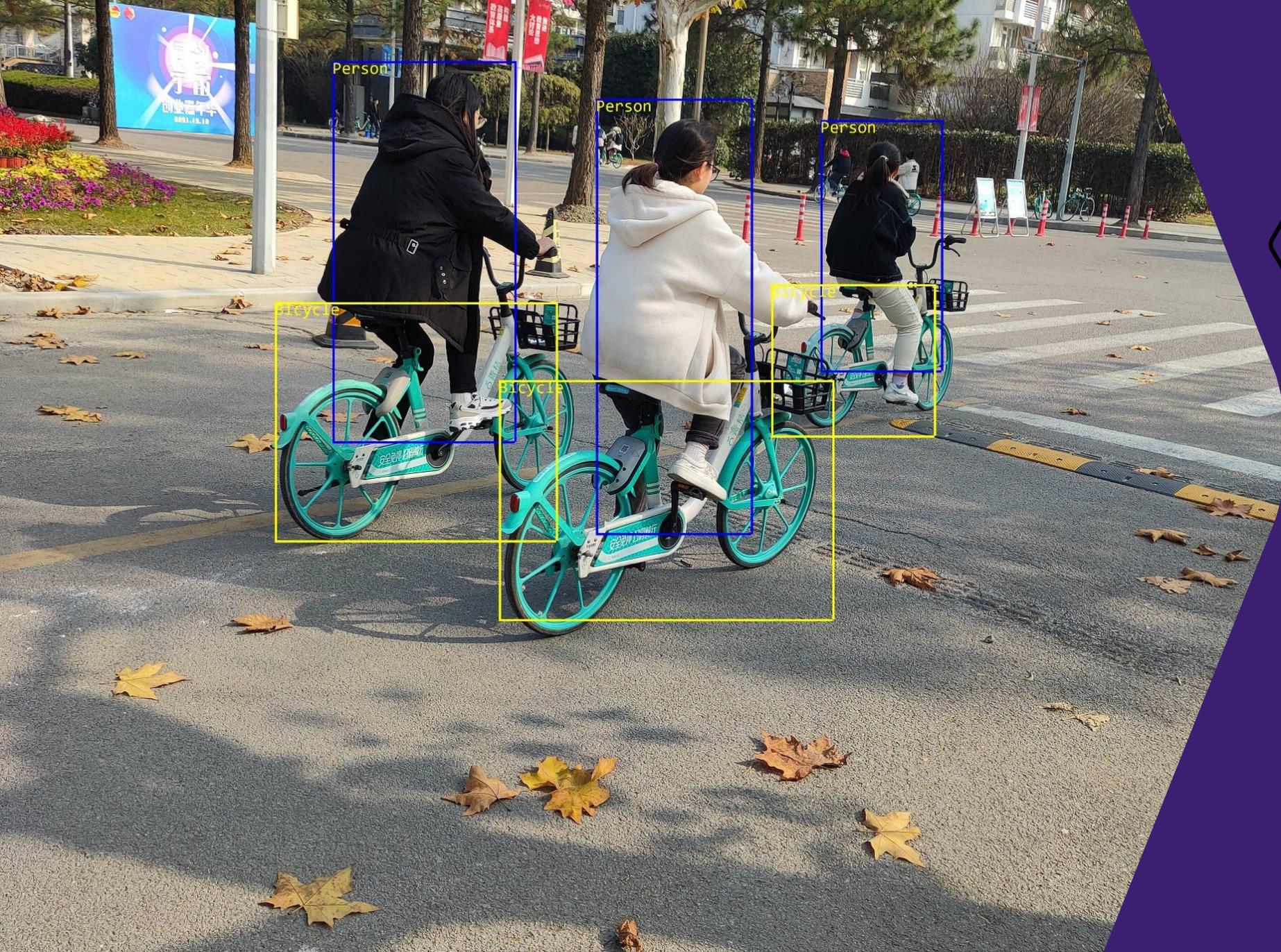
Person

Bicycle



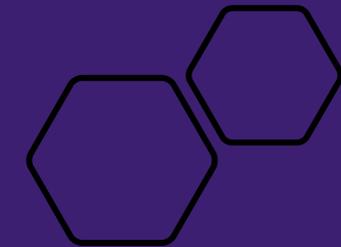
## 效果

自我感觉还不错。



## 效果

自我感觉还不错。



# 效果

自我感觉还不错。



# 声明



在本模型训练过程中，不存在任何形式的将挑战集放入到训练集的行为。  
特此声明。



## 项目分工

基本上每个人都进行了完整过程的学习，但仍然各有分工。

### 数据

- 数据采集
- 数据处理

### 训练

- 线上训练
- 本地训练（配置环境）
- 调参
- 优化

### 预测

- 预测优化
- 进一步优化思路



## 项目日志

- 
- 2021/12/14 户外拍摄
  - 2021/12/14 开组会，分工
  - 2021/12/14 标记数据
  - 2021/12/15 调参、线上训练
  - 2021/12/16 实验室配置环境
  - 2021/12/17 实验室配置环境、炼丹
  - 2021/12/17-2021/12/19 炼丹



数据



# 数据采集 全体

校园散步，非常欢乐。  
拍照+标注。



# 数据采集 全体

校园散步，非常欢乐。  
拍照+标注。





# 数据处理 成笑行



训练集 + 从几位学长处索取的一些数据 + 网上的  
开源数据

= 5275张

自己拍的照片 + 网上的开源汽车数据（朱泽凯）

= 940张





## 数据处理 成笑行



也是体力活。

写了一堆Python。

熟悉了 Python 文件库的用法。



# 数据处理

## 朱泽凯

标注文件转换

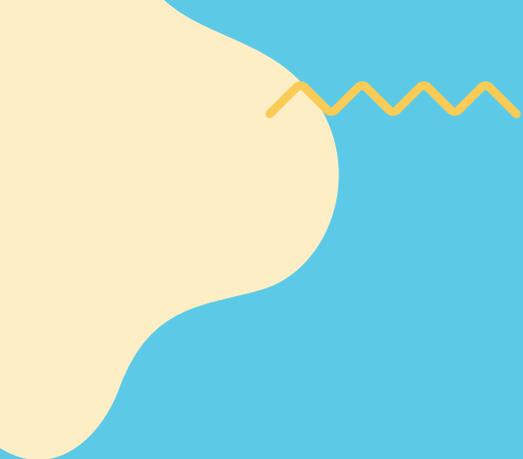
```
1.py
1 import sqlite3
2
3
4 db = sqlite3.connect('1.db', check_same_thread=False)
5 cdb = db.cursor()
6
7
8 cdb.execute("SELECT image, xmin, ymin, xmax, ymax FROM a")
9 ans = cdb.fetchall()
10
11 for row in ans:
12     print("%s\t{\'value\':\'car\',\'coordinate\':[[%s,%s],[%s,%s]]}" %row)
```

1.txt

```
1 vid_4_1000.jpg {"value": "car", "coordinate": [[281.2590449,187.0350708],[327.7279305,223.225547]]}
2 vid_4_1000.jpg {"value": "car", "coordinate": [[15.16353111,187.0350708],[120.3299566,236.4301802]]}
3 vid_4_10040.jpg {"value": "car", "coordinate": [[239.1924747,176.7648005],[361.9681621,236.4301802]]}
4 vid_4_10020.jpg {"value": "car", "coordinate": [[496.4833575,172.3632561],[630.0202605,231.5395753]]}
5 vid_4_10060.jpg {"value": "car", "coordinate": [[16.63096961,186.5460103],[132.5586107,238.3864221]]}
6 vid_4_10100.jpg {"value": "car", "coordinate": [[447.568741,160.6258044],[582.0839363,232.5176963]]}
7 vid_4_10120.jpg {"value": "car", "coordinate": [[168.7554269,180.6772844],[304.7380608,246.7004505]]}
8 vid_4_10140.jpg {"value": "car", "coordinate": [[0,188.9913127],[85.11143271,249.1457529]]}
9 vid_4_1020.jpg {"value": "car", "coordinate": [[202.5065123,189.4803732],[239.1924747,229.0942728]]}
10 vid_4_1040.jpg {"value": "car", "coordinate": [[116.4167873,189.9694337],[180.4949349,229.0942728]]}
11 vid_4_10480.jpg {"value": "car", "coordinate": [[473.9826339,172.8523166],[602.1389291,231.0505148]]}
12 vid_4_10500.jpg {"value": "car", "coordinate": [[149.6787265,190.4584942],[268.5412446,237.4083012]]}
13 vid_4_10520.jpg {"value": "car", "coordinate": [[0,195.3490991],[36.68596237,234.4739382]]}
14 vid_4_1060.jpg {"value": "car", "coordinate": [[45.49059334,192.9037967],[112.5036179,225.6708494]]}
15 vid_4_10960.jpg {"value": "car", "coordinate": [[614.8567294,175.297619],[676,217.3568211]]}
16 vid_4_10980.jpg {"value": "car", "coordinate": [[398.6541245,188.5022523],[518.0057887,236.9192407]]}
17 vid_4_11000.jpg {"value": "car", "coordinate": [[149.1895803,194.8600386],[265.6063676,242.0849421]]}
18 vid_4_11020.jpg {"value": "car", "coordinate": [[0,197.7944015],[75.81765557,237.4083012]]}
19 vid_4_11240.jpg {"value": "car", "coordinate": [[497.4616498,184.1007079],[633.9334298,231.0505148]]}
20 vid_4_11260.jpg {"value": "car", "coordinate": [[275.878437,197.3053411],[405.9913169,244.7442085]]}
21 vid_4_11280.jpg {"value": "car", "coordinate": [[62.61070912,201.217825],[186.853835,243.7660875]]}
22 vid_4_11380.jpg {"value": "car", "coordinate": [[514.5817656,176.27574],[615.8350217,229.0942728]]}
23 vid_4_11400.jpg {"value": "car", "coordinate": [[331.1519537,192.4147362],[429.959479,233.9848777]]}
24 vid_4_11420.jpg {"value": "car", "coordinate": [[137.9392185,197.3053411],[238.7033285,236.4301802]]}
25 vid_4_11440.jpg {"value": "car", "coordinate": [[0,199.261583],[74.35021708,235.9411197]]}
```

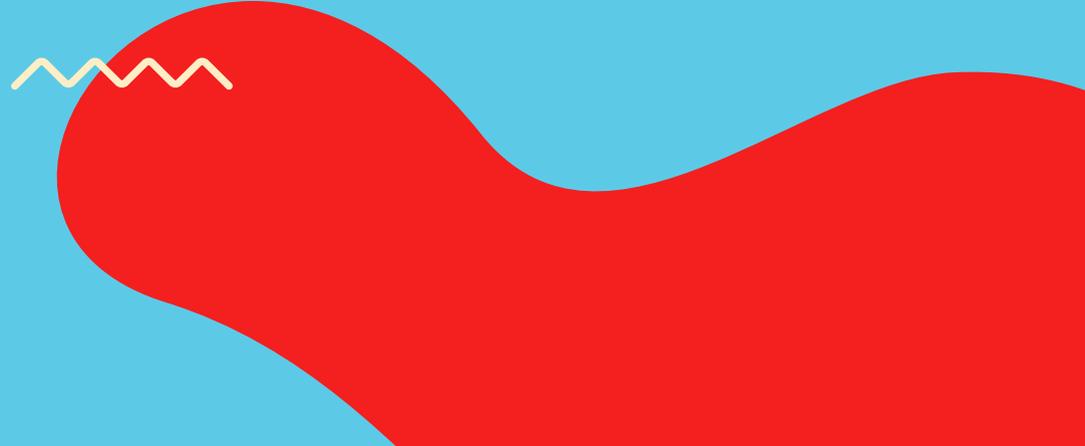
1.csv

```
1 image,xmin,ymin,xmax,ymax
2 vid_4_1000.jpg,281.2590449,187.0350708,327.7279305,223.225547
3 vid_4_10000.jpg,15.16353111,187.0350708,120.3299566,236.4301802
4 vid_4_10040.jpg,239.1924747,176.7648005,361.9681621,236.4301802
5 vid_4_10020.jpg,496.4833575,172.3632561,630.0202605,231.5395753
6 vid_4_10060.jpg,16.63096961,186.5460103,132.5586107,238.3864221
7 vid_4_10100.jpg,447.568741,160.6258044,582.0839363,232.5176963
8 vid_4_10120.jpg,168.7554269,180.6772844,304.7380608,246.7004505
9 vid_4_10140.jpg,0,188.9913127,85.11143271,249.1457529
10 vid_4_1020.jpg,202.5065123,189.4803732,239.1924747,229.0942728
11 vid_4_1040.jpg,116.4167873,189.9694337,180.4949349,229.0942728
12 vid_4_10480.jpg,473.9826339,172.8523166,602.1389291,231.0505148
13 vid_4_10500.jpg,149.6787265,190.4584942,268.5412446,237.4083012
14 vid_4_10520.jpg,0,195.3490991,36.68596237,234.4739382
15 vid_4_1060.jpg,45.49059334,192.9037967,112.5036179,225.6708494
16 vid_4_10960.jpg,614.8567294,175.297619,676,217.3568211
17 vid_4_10980.jpg,398.6541245,188.5022523,518.0057887,236.9192407
18 vid_4_11000.jpg,149.1895803,194.8600386,265.6063676,242.0849421
19 vid_4_11020.jpg,0,197.7944015,75.81765557,237.4083012
20 vid_4_11240.jpg,497.4616498,184.1007079,633.9334298,231.0505148
21 vid_4_11260.jpg,275.878437,197.3053411,405.9913169,244.7442085
22 vid_4_11280.jpg,62.61070912,201.217825,186.853835,243.7660875
23 vid_4_11380.jpg,514.5817656,176.27574,615.8350217,229.0942728
24 vid_4_11400.jpg,331.1519537,192.4147362,429.959479,233.9848777
25 vid_4_11420.jpg,137.9392185,197.3053411,238.7033285,236.4301802
```



---

训练





## 线上训练

AIStudio是世界上最好用的环境！  
跑了一天就没时间了。  
传文件还有 150MB 限制。  
一晚上能跑 50 轮，还算不错。





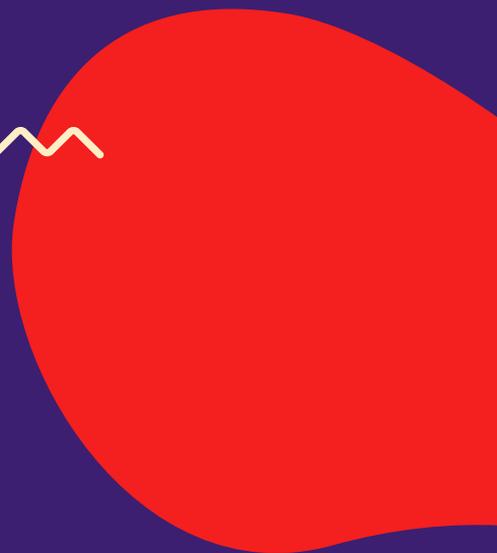
# 线上训练

刚开始的效果并不好。



# 线上训练

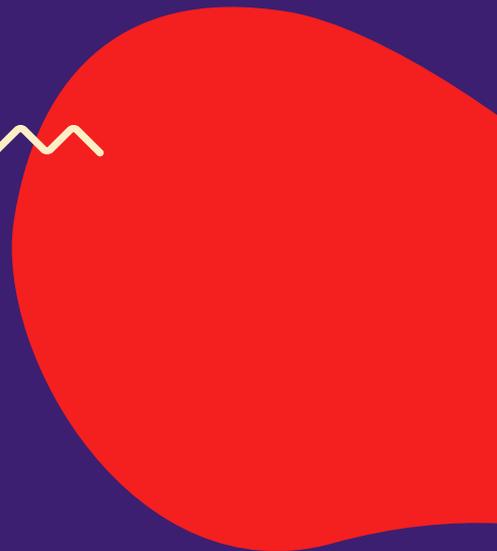
第二次。



# 线上训练



# 线上训练





## 本地训练

感谢龚晓亮老师。

9900K + 2080Ti 真好用。

原来配一天环境是真的有可能发生的。

不过摸索出了本地配置的文档。

最后上传的是成笑行的模型，但王钧涛、张弛、庞皓天、李博宇都各自进行了训练，并发现了一些优化和改进方法。

最后提交的模型跑了对所有数据跑了约300轮。





# 本地训练

最后总共迭代了约10个版本。

改进：

数据的增加和处理筛选。

调参。

调整预测。

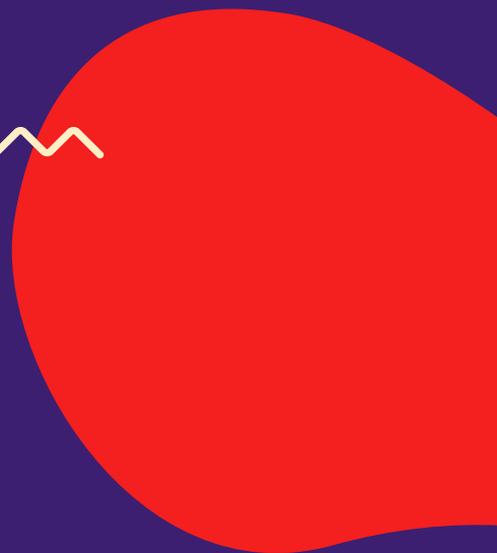
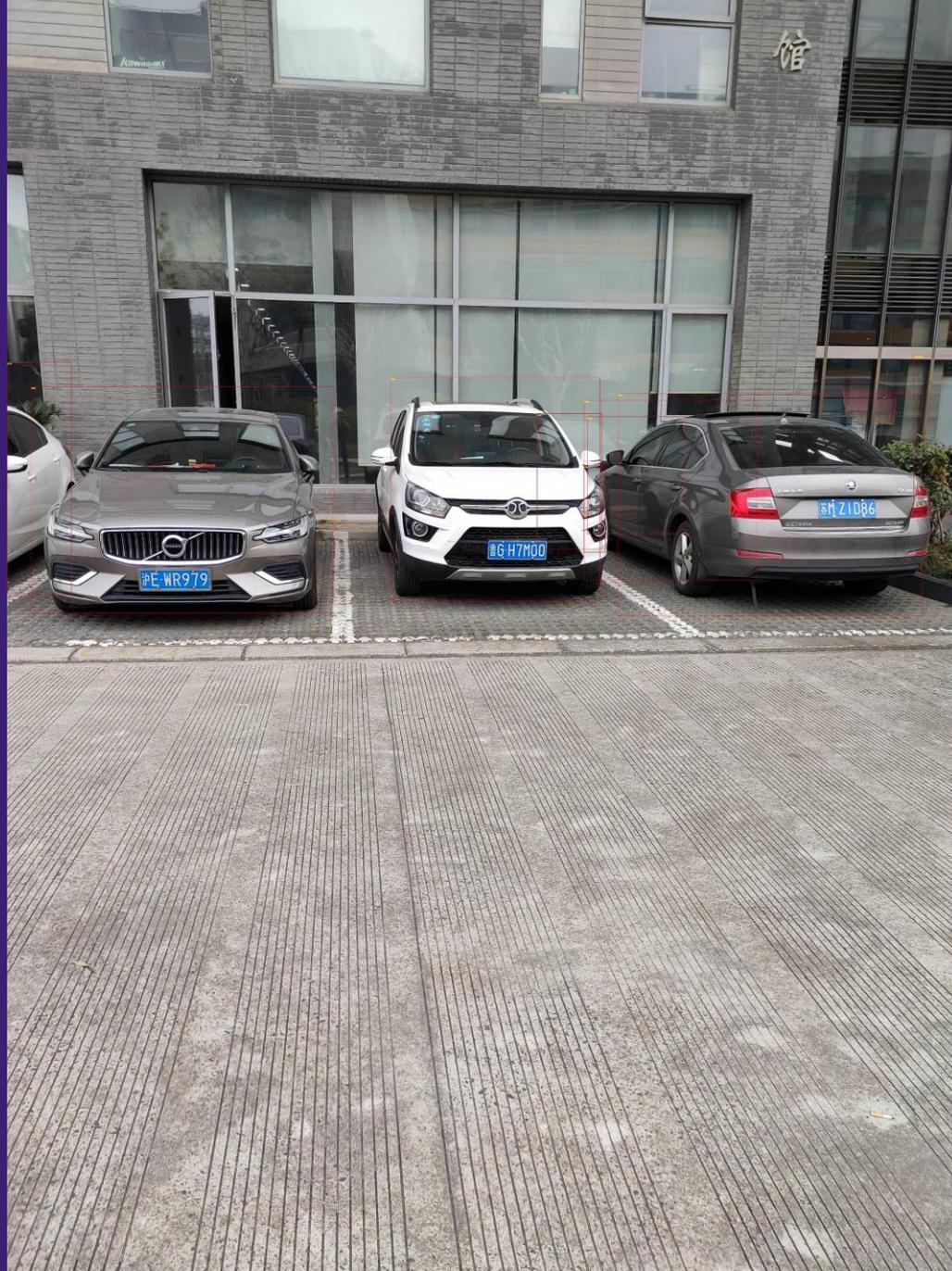


# 本地训练



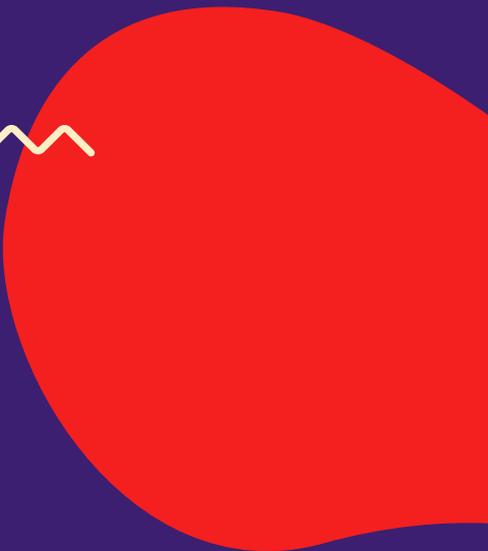
# 本地训练

第3次。



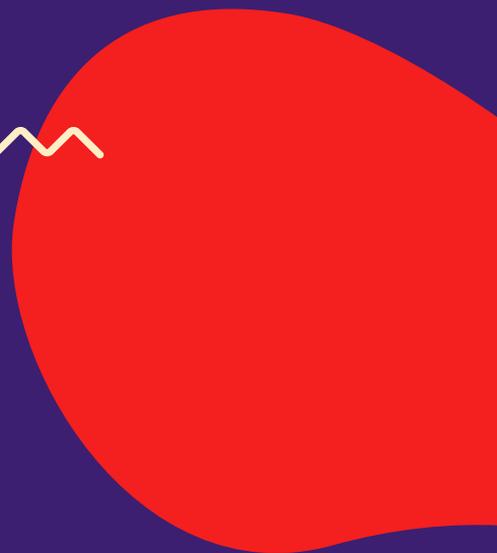
# 本地训练

第2次。



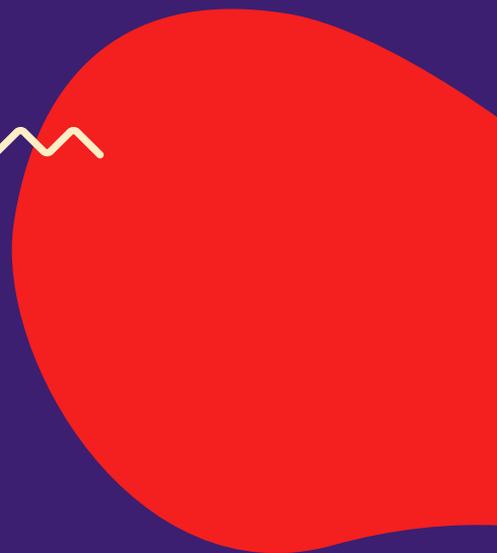
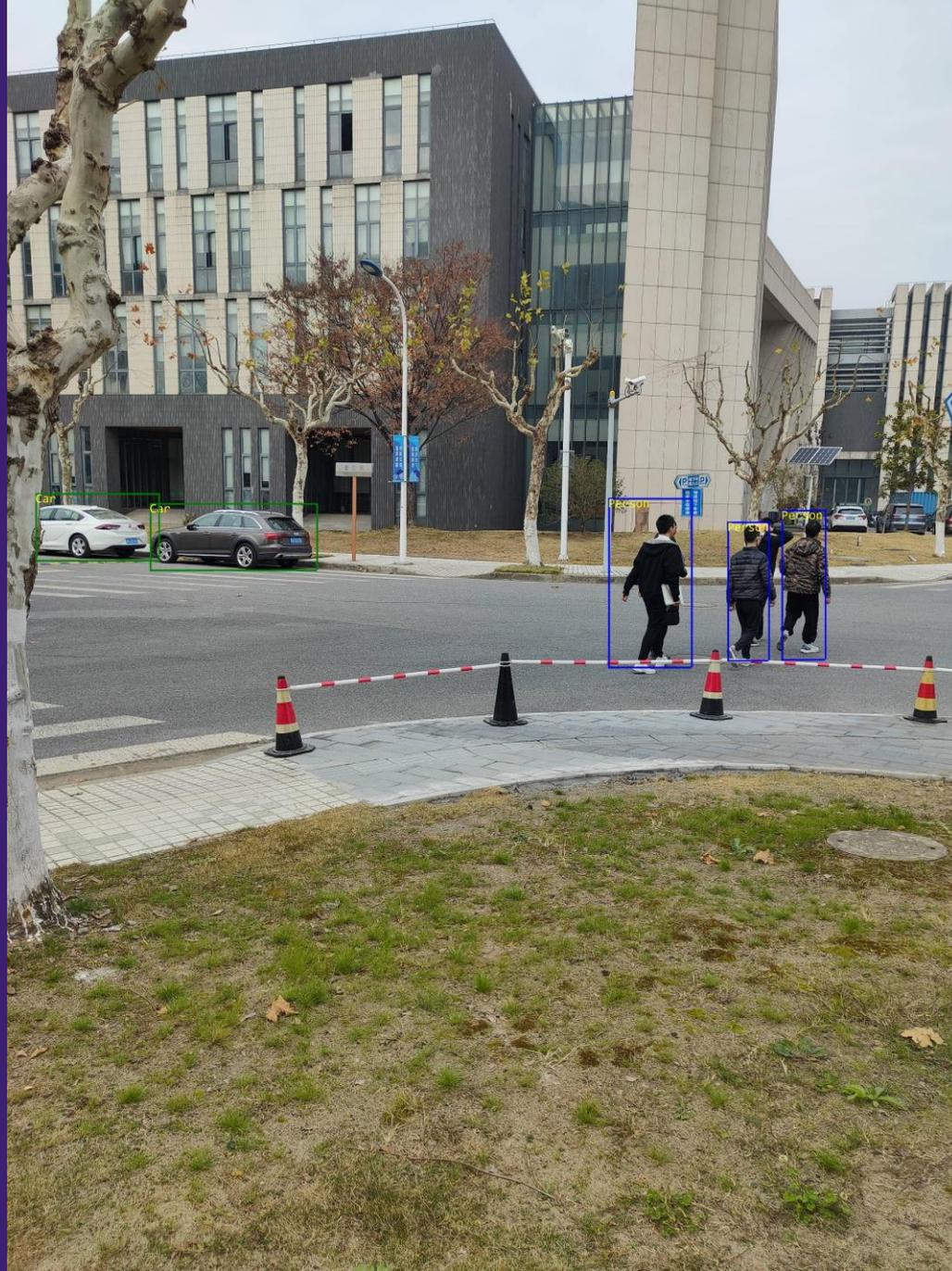
# 本地训练

第 $n$  ( $n \approx 7$ ) 次。



# 本地训练

第 $n$  ( $n \approx 10$ ) 次。





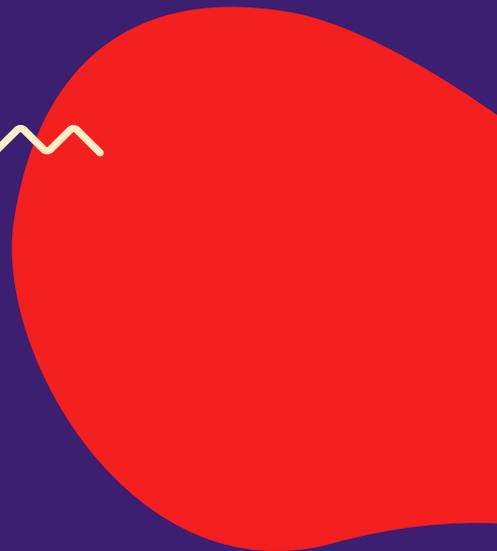
## 一些尝试 - 王钧涛

尝试寻找一些开源数据集；

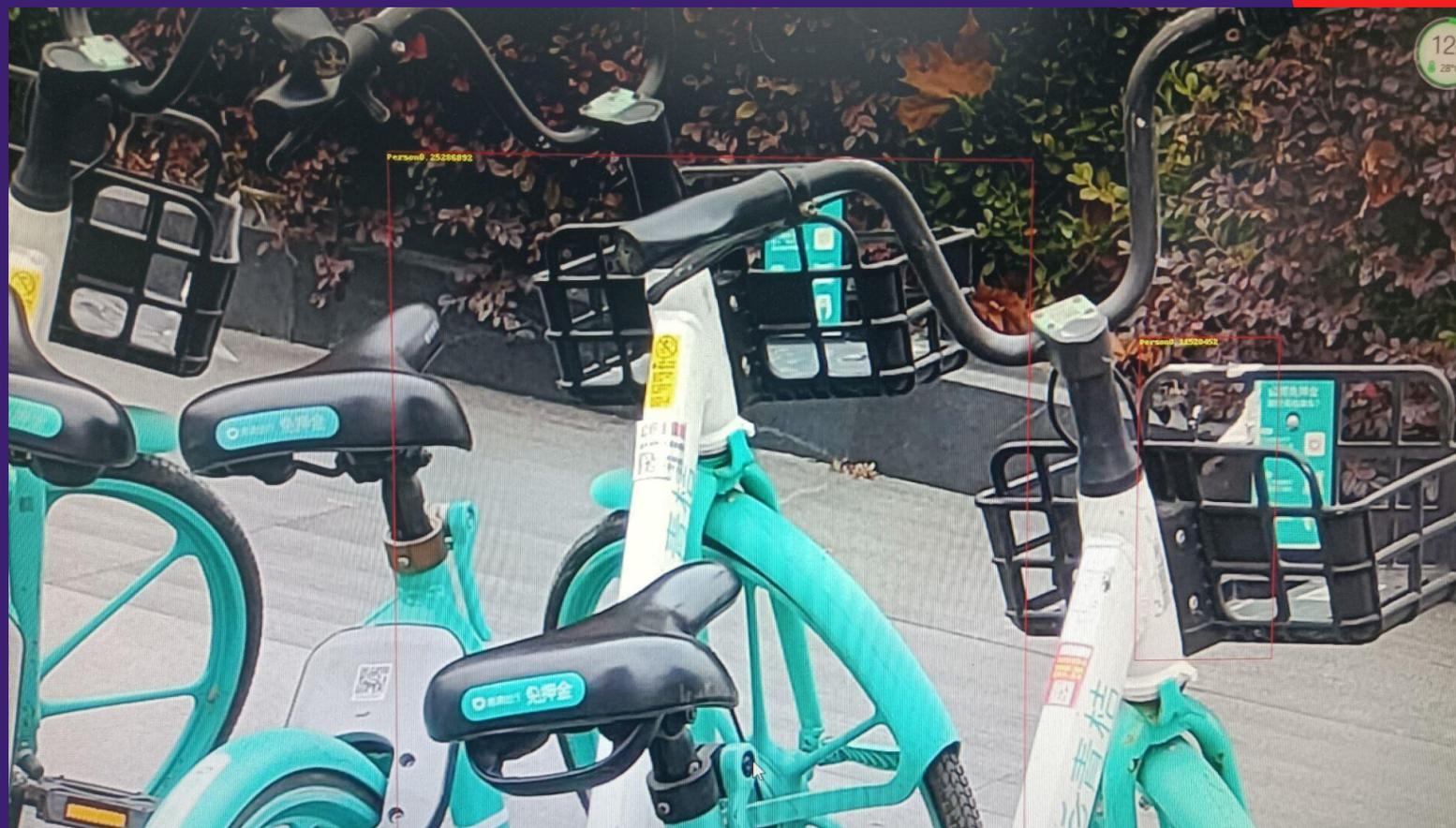
提取了PASCAL VOC2012中的相关数据放在成笑行优化后的模型中跑；

发现开源数据集对于车辆的效果有提升（近乎完美），但是对行人和自行车（尤其是校园内大量的共享单车）的效果不佳，主要表现在自行车基本不能识别和汽车只能识别一辆；

最终提取了很小一部分数据集放入训练集中。



# 一些尝试 - 王钧涛



## 一些尝试 - 张弛

飞桨平台的包版本较低，因此尝试了更新的版本

paddlepaddle 2.2

python 3.9.8

CUDA 10.2

CUDNN 7.6.5

使用较低版本驱动，batch-size取到6已经占满显存

使用高版本驱动，batch-size可以取到16，显存10/11GB

提高了训练效率



# 一些尝试 - 陈奕澄

在本地配置环境  
标记收集的图片





## 一些尝试 - 庞皓天

在本地配置环境

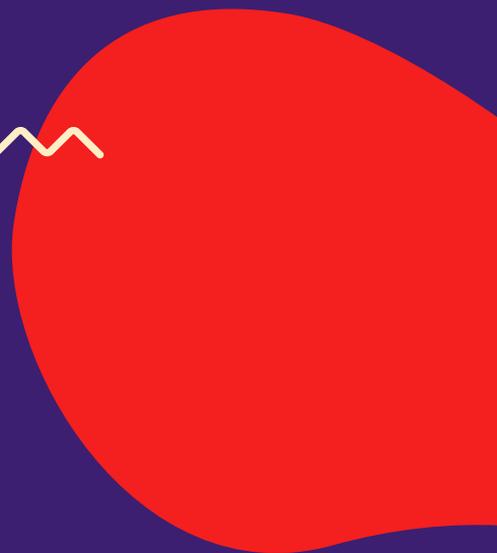
标记收集的图片

尝试跑自己的一个模型，跑了接近二百轮

对收集的不同的数据集进行训练，找出较为合适的部分

尝试修正函数的参数，找出比较有潜力的方向

帮助检查和验收其他参数的模型





# 一些尝试 - 李博宇

配置环境  
by快写





## 一些尝试 - 李博宇

微调模型参数

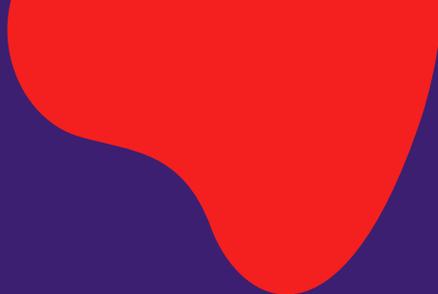
对现有模型迭代训练，避免每次从头训练的繁冗

尝试改进图片预处理方式

后续的模式修正很重要！



(GPU内存又双叒叕炸了。。。苦)



## 调参

懂的都懂。

Anchor 非常重要!

batch\_size

nms\_thresh

Ignore\_thresh

valid\_thresh

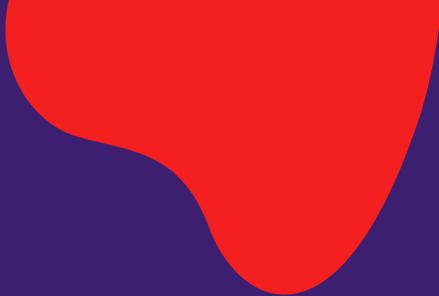
优化显存。

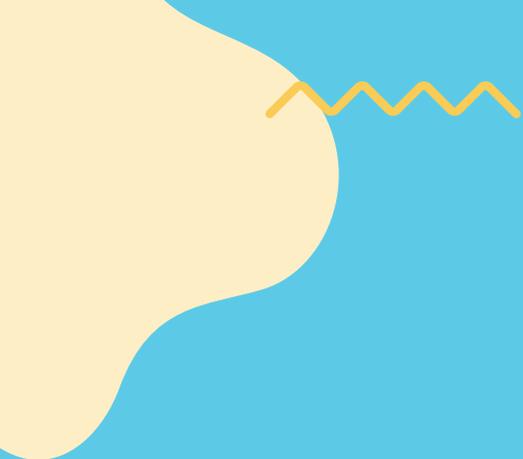


## 优化



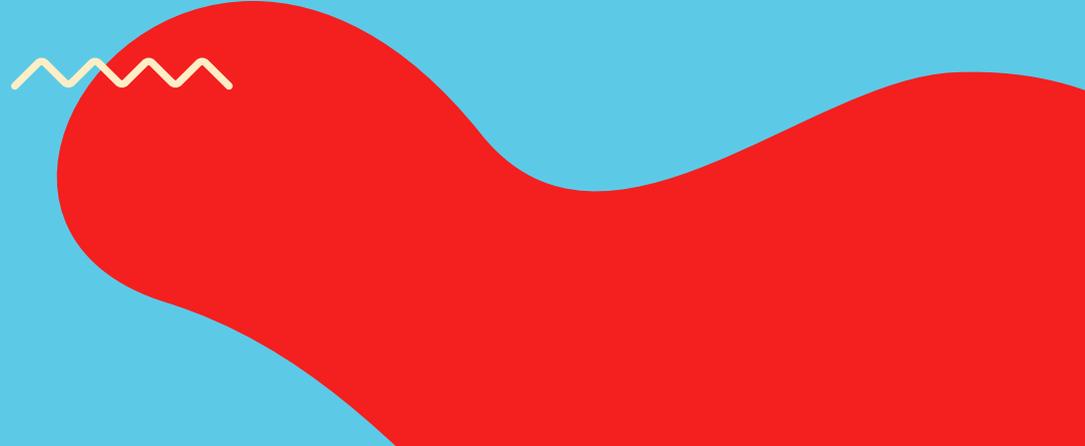
刚才应该讲过了。  
可以试试更改优化器为Adam。  
经过和同学讨论，没用采用计算mAP和准确率的代码。





---

预测





## 优化

“

计算每个预测值的置信度，只显示置信度比较高的预测结果，可以有效减少有很多框的情况。





## 进一步改进的思路

将YOLOv3骨架网络更换为ResNet50-VD。  
引入可变形卷积替代原始卷积操作。  
在FPN部分增加DropBlock模块，提高模型泛化能力。



## 缺陷

在和训练集相差比较大的图片上效果不太好。

有时候Person会认错。



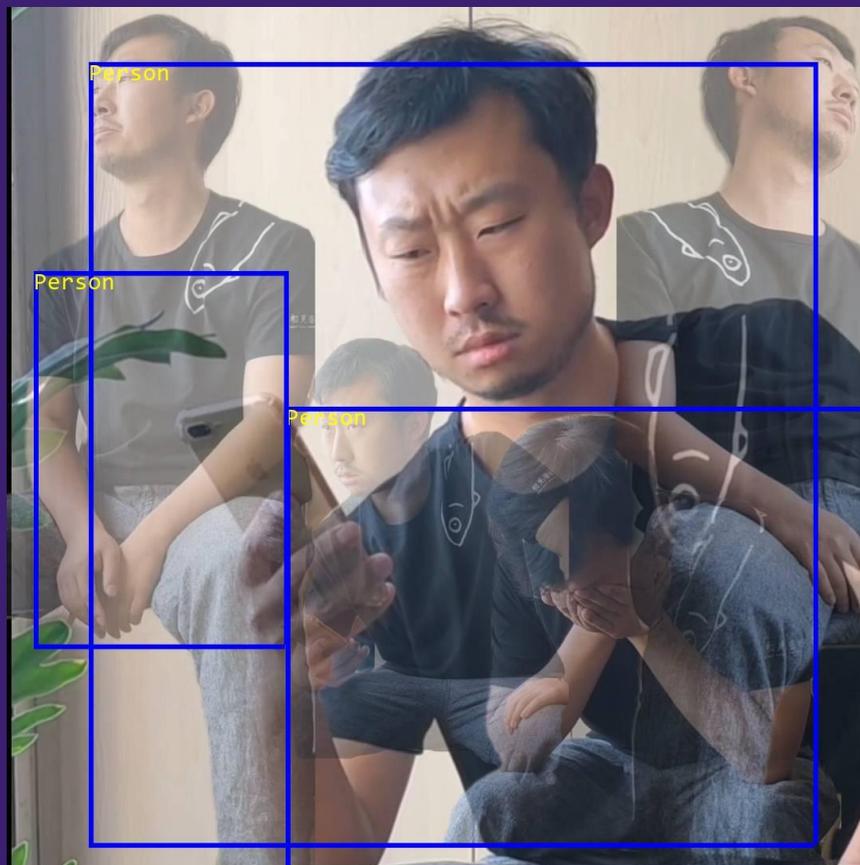
试试其他图片?



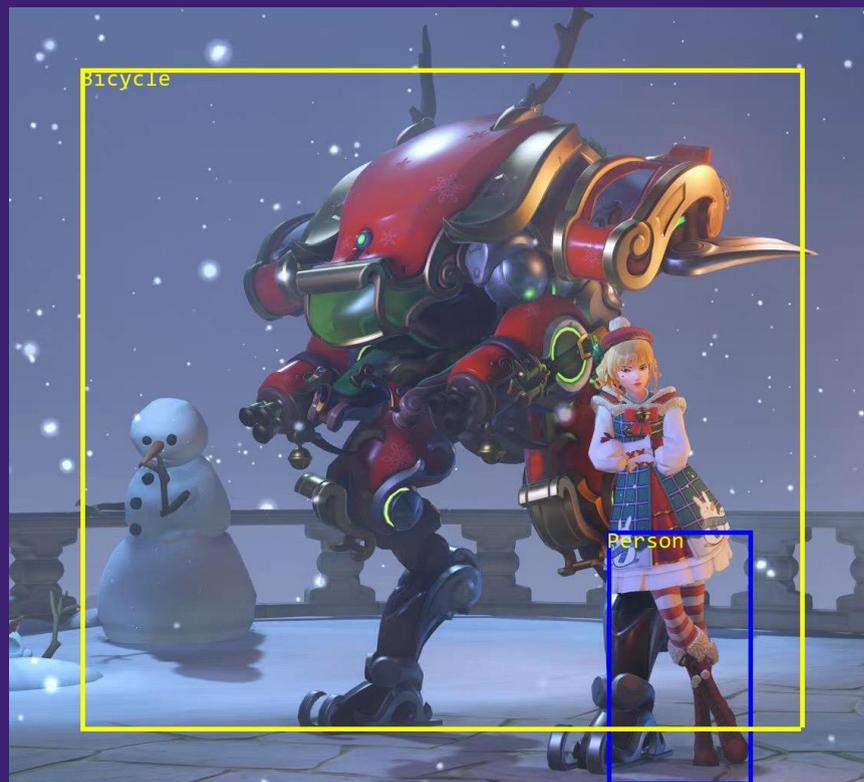
试试其他图片?



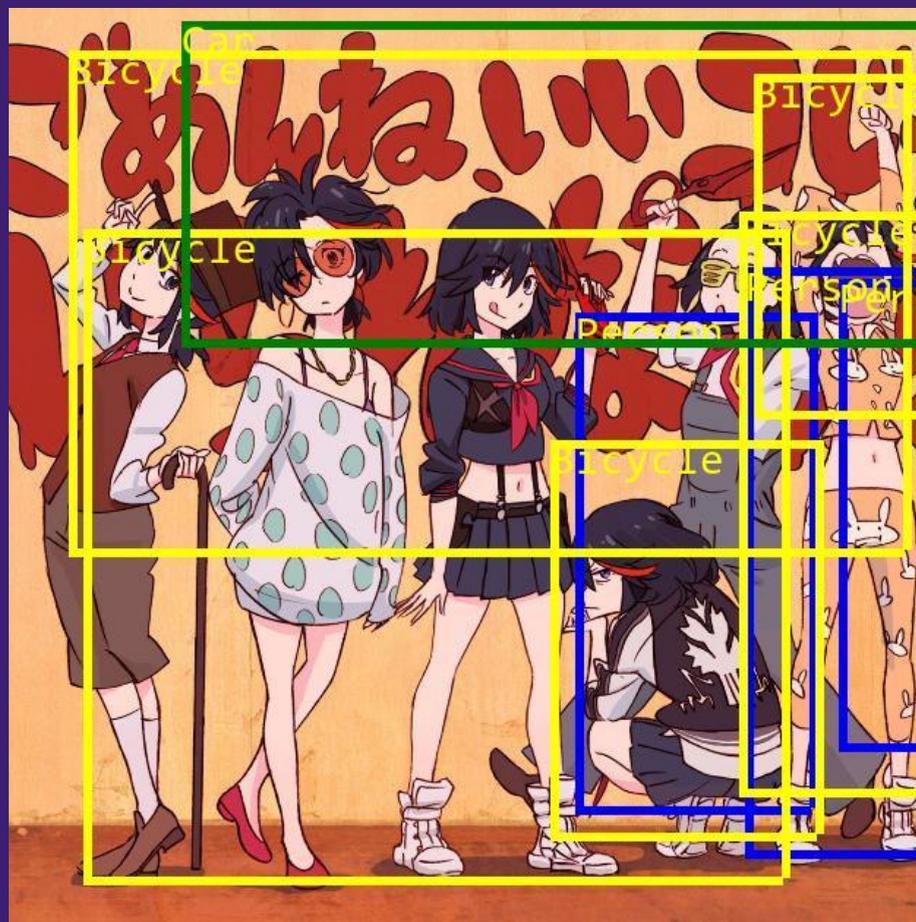
试试其他图片?



试试其他图片?

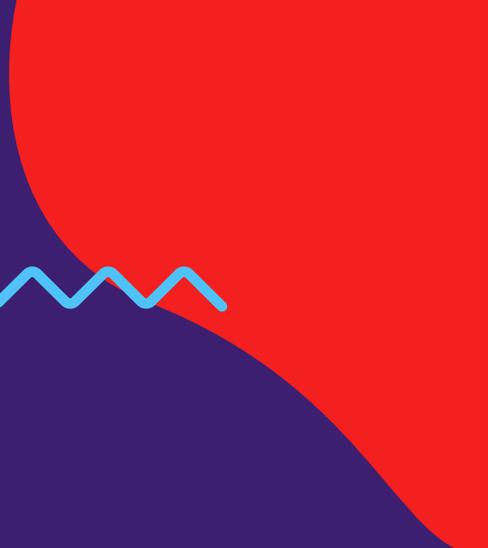


试试其他图片?





## 参考文献

- AIStudio官方文档
  - Paddlepaddle文档
  - 往届学长的演示PPT
  - Google
  - StackOverFlow
  - 其他博客、文档
- 
- 
- 
- 



谢谢欣赏