

# 一个CV算法工程师的小反思

计算机视觉联盟 5 days ago



计算机视觉联盟

团队成员为国内各大高校博士，专注于深度学习、机器学习、图像解译、人工智能、无...  
114篇原创内容

Official Account

点上方 [计算机视觉联盟](#) 获取更多干货

仅作学术分享，不代表本公众号立场，侵权联系删除

转载于：作者 | 努力的伍六七@知乎（已授权）

来源 | <https://zhuanlan.zhihu.com/p/363354912>

编辑 | 极市平台

## AI博士笔记系列推荐

[周志华《机器学习》手推笔记正式开源！可打印版本附pdf下载链接](#)

大佬可以写总结给别人指导，菜鸟可以写总结给别人指坑。

原本打算是正式工作满一年以后写的，最近反思了很多事情，也找到了很多不足之处。怕以后忘记了，就想到什么写什么，什么没想到以后就补上。

## 1. 算法篇

- 虽然我没有发表过顶会论文，谈不上学术成就。但是我还是要硬谈，在实际我们做算法落地的时候，最重要的东西确实是数据。一个CV算法项目的流程迭代基本都是：产品分析，业务指标制定，算法调研，采集数据1W左右就够了，出一个demo，产品测试，有初步bad case，其中肯定有泛化性问题。因为就1W数据。先增加数据量级，在增大数据量级的时候要考虑badcase有哪些情况，还有数据采集这个事情也是需要一定经验的。当你考虑的case比较多的时候，很难做到穷举。制定好数据采集策略非常重要。
- 算法优化的套路不是提高指标，是要解决业务问题。学术上统计一个测试集的mAP(比如检测)，业务中是要根据实际业务场景分开简历测试集，比如暗光下测试集，带遮挡测试集，通用场景测试集等等，要针对业务场景进行优化，业务场景下测试集就是算法和测试那边沟通。

- 端侧的算法优化要考虑部署问题，就这么说吧，你打开SNPE的文档，基本照着里面的支持看基本就是八九不离十了。有些常见的考虑点时激活函数基本就是只用Relu，带有逻辑操作的比如ROIAlign等等，上采样的方式有Deconv, Interp(nearest, bilinear), UpPooling, 空洞卷积这几种，但是不同的方式在不同的芯片平台支持是不一样的。还有全连接的耗时可能在有些平台上也不太行。简单来说除了conv relu，其他的东西都要在做之前有个调研和考虑。还有可能会有限制输出Tensor数量和输入Tensor数量的问题。其他的就是不同平台的量化支持不同了。讲了这些我想现在还有人在用VGG还是可以理解的。Make VGG Great Again.
- 工作中最重要的不是论文的创新，尤其在初期做算法的套路，流程要清楚。以及执行力要Max，还要有一定的数据敏感性。在标注数据的时候需要界定什么是正样本什么是负样本，什么是无法判断(ignore)，什么时候需要脑补，什么时候不需要脑补，这个非常非常重要，数据标错了后续要花费很多力气改正。
- 要保证实验的正确性，一开始实习的时候回经常跑错实验，或者实验跑到一半挂了，这会大大降低效率。做事情要慢一点，但是要正确。
- 用tmux管理实验非常方便，要写好实验日志，保证自己能够看到哪一行就知道这个实验做什么改动，并且可以直接找到实验结果保存的地方，我一般通过名字来体现，比较复杂的就写备注。
- CV不像推荐广告NLP，不需要用到SQL，Hadoop之类的数据查询等工具，CV数据的管理要有自己的一套章法。要做到每一张图片都是可溯源的，图片和视频，抽帧频率，这些都需要列表。如果需要对图片做后处理，比如人脸数据，中间的结果人脸检测bbox，抠出来的图等等都要有记录，脚本要整理好，一个任务从抽帧开始到最后训练的图片，分步骤处理并且每个步骤都保留中间结果，主要不要保留图片，保留坐标等信息。保留图片会占用太多存储空间。每一个数据都要有专门的表格来记录，不然以后整理其数据来有你难受的。
- 因为图片和标注都是放在磁盘上，所以要有自己的风格，我的风格就是data目录放视频，images目录放图片，Scripts目录放脚本，Ann目录放处理完的标注文件，Ann\_pre放标注人员返回的原始标注文件。其中每个目录中会根据任务需求建立子目录，数据的信息必须要包括的是任务+时间，其他看着需求加。
- 图片和标注文件定好之后加下只读权限，不然哪天手贱给删了不知道上哪哭去。
- 训练模型之前要确认的事情有:进网络的图片和GT画出来，确认是否会有问题。网络的Target画出来确认是否会有问题，这两点没问题Loss再下降那恭喜你等着看demo了
- WarmUp要加上，这东西谁用都说好
- GradNorm打出来，如果崩了也好分析

- LearningRate一般就SGD和Adam，CosineLR等这些不怎么影响你最终的结果，可能会加快收敛。
- 要积累自己的代码库，这样很多时候只要搜一下copy过来改改就可以用，最重要的是不用再去查API浪费时间了
- 提升代码能力和算法能力的最好的方式就是看好的开源代码，多看多写才能提高思维能力。
- CV算法好卷啊，检测，分割，识别，跟踪，OCR，ReID，分类，GAN，动作迁移，SLAM，深度估计，3D相关，姿态估计，图像检索，NAS，量化，立体匹配，图像复原，.....，What's Fuck，我这一写感觉自己跟个白痴一样好多都不懂。。。。
- 框架上要加的功能有Multi-Task训练(不同卡上运行不同Task|根据Task的标签手动切割网路输出计算Loss)，模型并行，数据并行，多数据源训练(不同数据可以用不同数据增强)，多卡并行，转模型，FP16训练，量化训练，demo可视化，BadCase分析，多测试集评估，NAS搜索，跨数据集训练(A数据中有人体，B数据也有人体，但是标注的label不同，需要制定一套方案把不同的数据集中的标签重定义)。其中有些不是必要的就不加
- 要用好vim和shell,在CV任务里面vim最常用的是一些文件的合并，交集，差集，排序，去重等等。shell命令+管道机制可以让你快速地做很多事情，不然从头开始写python代码也是比较费时间的。最常用的是find + 通配符找文件，然后对文件进行删除，移动等等操作
- opencv是必须要熟悉的，简单的读取，存储，画图的API是必须记住。其他的要求积累脚本。不过目前来看很少用到一些传统特征。如果需要用到那句把这些脚本都存储下来。
- 数据标注完成后处理成数据集一定要考虑仔细了，在转格式的时候一定要加上一些格式确认，比如标注检测框是否有左上角的坐标值却小于右下角这种情况，这个图片是否是已经被损坏了等等。很多时候你找半天bug发现是数据的问题。
- 工作中没人关注你的算法是不是新的，是不是牛逼，代码有多难写，不能为了新颖而新颖。能解决badcase的就是好优化，不然就是白搭。
- 要保证自己实验的可复现性，做优化的过程中会经常需要改一些小细节。可能对之前的实验有影响，尽量改成传参的方式，不然也可以继承重新写一个类。
- 算法工程师要多向工程部署的同事请教，看看模型部署的时候需要考虑的东西有哪些，资源占用，速度优化等等是怎么做的。
- 不要忽略算法的后处理，在实际用的时候经常会加一些后处理的逻辑判断，这个也非常重要，有时候能解决很多问题。

- 目前我知道算法优化的有效常用方案是：加数据，multisource训练，数据增强，好的pretrain\_model，正负样本均衡的优化，正负样本采样相关优化，知识蒸馏，额外监督比如检测中加分割基本能涨点，attention模块，多尺度，特征融合，合适的backbone，NAS搜索(应该大家都在做)，量化训练，剪枝。其他比如算法流程的大改动，一般人真做不了，比如VGG->MobileNet,DeepSort->CenterTrack，YOLO->YOLOv5，这些硬通货都是大神的杰作，要向大神学习，但是不要一口吃胖子。
- python脚本要熟练编写，多线程操作要信手拈来，数据处理经常会需要多线程处理

## 2.成长篇

- 要相信自己，不要总觉得自己不如别人，要对自己有信心，才能让别人对你有信心。
- 要不断学习，我的人生宗旨就是我可以菜得被开除，但是我不能因为无所事事浑水摸鱼被嫌弃。
- 要和同事友好相处，工作之后明显就交际圈小了很多，同事是交流最多的人，不管怎么样，都要客客气气。多帮别人一点，别人才会多帮你一点。
- 要拼搏，但是不要拼命，程序员本质上也只是一个打工的，干点活，拿点钱，不要把命搭上去。生活 还会很美好的
- 要去了解外界，不要一头就在代码里，要有个爱好，跳舞，跑步，拍照，旅游，这样你才会能找到生活的感觉。
- 心态要躺平，我的心态就是什么事情我都会努力去做，如果我努力去做却没能做好那是我能力问题，给我3.25，给我开除我也接受。我要做的就是总结下为什么没有把事情做好，去反思自己遇到的哪些问题，走了哪些弯路，下次不要再犯。
- 不要抱怨家庭的出生，抱怨永远解决不了问题，已经五六十岁的父母难道还指望他们改变你的命运让你成为富二代？

暂时先写到这里。。。

不管以后是不是还在干程序员，我都会一直坚持学习，坚持努力，还要坚持输出。

-----  
**END**  
-----

我是王博Kings，985AI博士，华为云专家、CSDN博客专家（人工智能领域优质作者）。单个AI开源项目现在已经获得了2100+标星。现在在做AI相关内容，欢迎一起交流学习、生活各方面的问题，一起加油进步！