发件人： **nick.xue** nick.xue@jian24.com
主题： Re: 周报_20180311
日期： 2018年3月14日 02:21
收件人： 席佳 jia.xi@jian24.com

辛苦写了这么多!

——————— Original Message ———————

Sender: 席佳<jia.xi@jian24.com>
Recipient: 薛文植<nick.xue@jian24.com>
Date: Monday, Mar 12, 2018 10:39
Subject: 周报_20180311

本周主要是想加快整个模块的处理速度 以及 修改测试的程序
1. 修改训练程序 因为之前的模型要用imagenet的pretrain 所以加载到全连接之前的参数要一致
问题是，前面一层的特征为512个float的数字，这样的话，如果想之后改成特征比对的话 耗时会比较大，所以加了一层bottlenet到128维数据 以后方便提特征比较
但是训练比较麻烦，没有了pretrain，要重新调整学习率 然后不断的重新训练才行 而且训练需要较久的时间

2.增加了一个前处理QA函数
1）比例不协调的图像不加入输入
2）用最快的方法找出相似的图片
3）相似图片里面选质量最好的一张留下到要输入的图片list里
0-0 找到了最快的比较方法 先求hsv-》h空间直方图-》后一个比较相似度-》加入list再比较
时间用时不等 20~500ms 看文件夹图片数量以及图片直接相似程度
在挑选出来的list里面再进行识别就会变快
测试了一把和原来比较发现识别率下降了20%... 有些相似的归为一类了 所以原本有些能识别的就没有进入list 所以剩下那个不一定能识别出来 所以下降了
所以将QA函数只剩下1），质量好的也不能留，因为一般情况下拿东西的都是糊的，不拿东西的才是清楚的，所以0-0 质量模块也不能单独使用

3. 测试模块修改
1）因为之前测试的写的比较乱 所以修改得规范 以便测试组使用
原来筛选图片全部选中再拉进terminal存成txt的模式变成了-》把正确图片拖入文件夹即可
也就是修改了得到的事件集 全部归好类以后 在那个事件生成一个right的文件夹就可以拖进去
然后我把这个程序生成exe了 所以window也可以直接用

2）修改了测试程序 0-0 原来挤在一起 现在改成一个一个函数了
重新定义了下三个评价函数：
a. 总的准确率： （Over_AC）
events_return_NULL + rightRetrun

Over_ACC= ----------

all_events

事件集本身没有这个对的图片，并且我们返回的NULL 以及事件集正确叫events_return_NULL，我们也返回正确图片的rightRetrun加在一起就是所有的正确的 比上 所有的事件集 得到的就是一个现场我们真正的正确率

b. 真实正确率（nn_AC）

rightReturn

nn_AC =----------

all_events - events_Null

就是正确的事件集和正确的返回rightReturn比上所有的有效数据（所有的事件集all_events - 无效的事件集（那些本身就没有正确图片的事件集））就是真实的正确率

c. 非null的正确率（our_nn_AC）

rightReturn + returnNull

our_nn_AC = ----------

all_events - events_Null

这个就是将returnnull也算作正确的情况下（就是看有没有错误的判断）的正确率

3)讲原来生成的文字模式改成了表格 更清晰易见

shelf	AC	nn_ac	our_nn_acc	
X08	0.866667	0.555556	0.888888889	
X01	0.666667	0.666667	0.666666667	
X20	0.5	0.5	1	
X09	0.942529	0.5	0.9	
X21	0.722222	0.6	1	
X17	0.5	0.333333	1	
X05	0.869048	0.625	1	
X18	0.714286	0.642857	0.928571429	
X22	0.538462	0.590909	0.954545455	
X06	0.875	0.545455	0.909090909	
X07	0.804878	0.615385	0.923076923	
X16	0.5	0.5	0.95	
X03	0.428571	0.423077	0.923076923	
X19	0.733333	0.727273	1	
X04	0.4	0.4	0.8	
X02	0.7	0.25	1	
sumall	0.786618	0.532258	0.930107527	
ight_even	urn_events	_events_nu	final_sum_bar_events_all	final_return_null_all
99	336	367	553	74

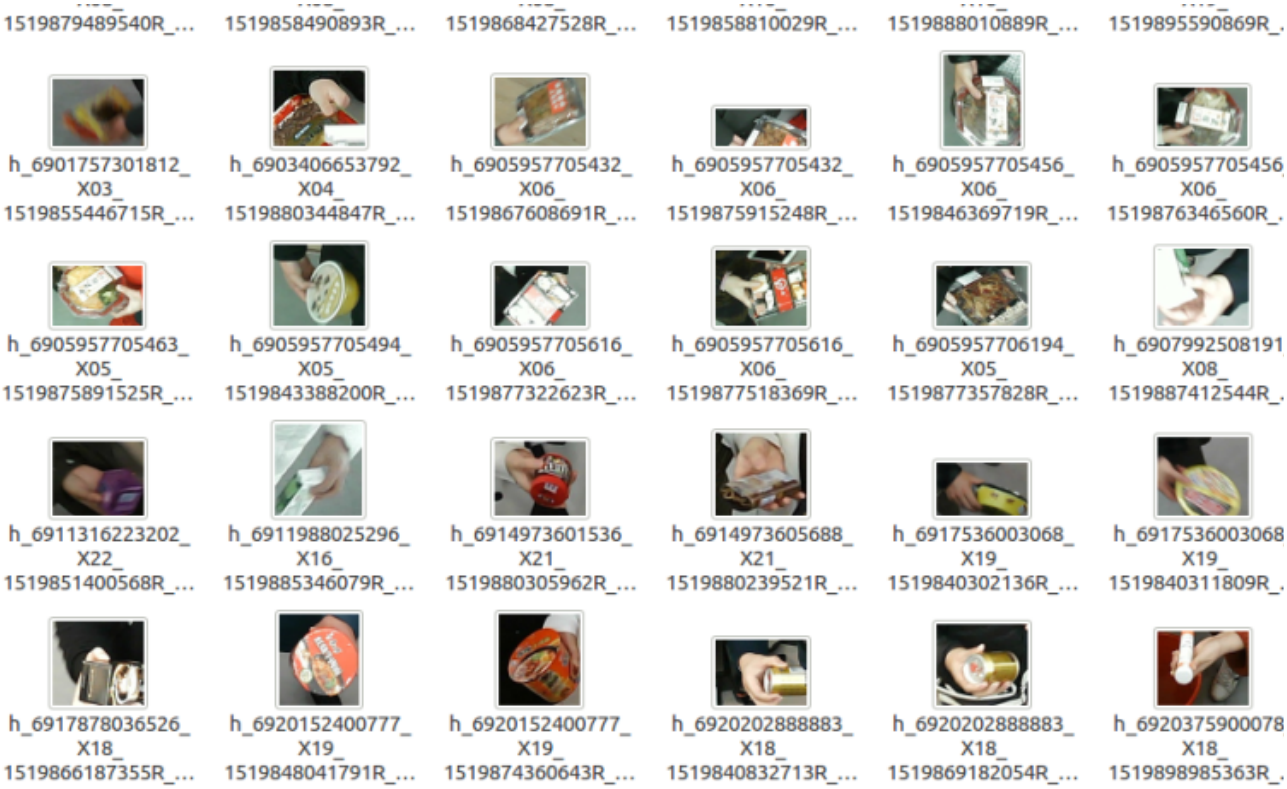
输出如上

但是有可能因为我们某些正确的图片没被我们加入正确txt（遗漏）

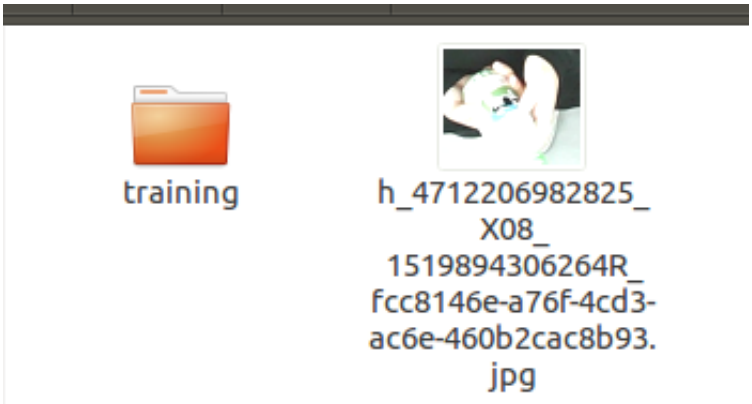
所以正确率是要再高一点的：

我在最后也将错误图片 返回null的 以及 正确的都输出 也包括每个货架每个商品的识别率

正确找出的图片一般张这样：



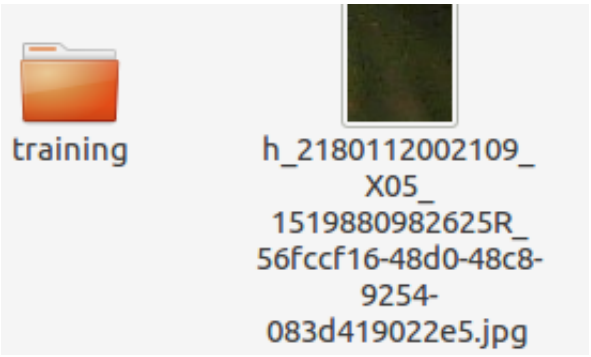
假的错误有：



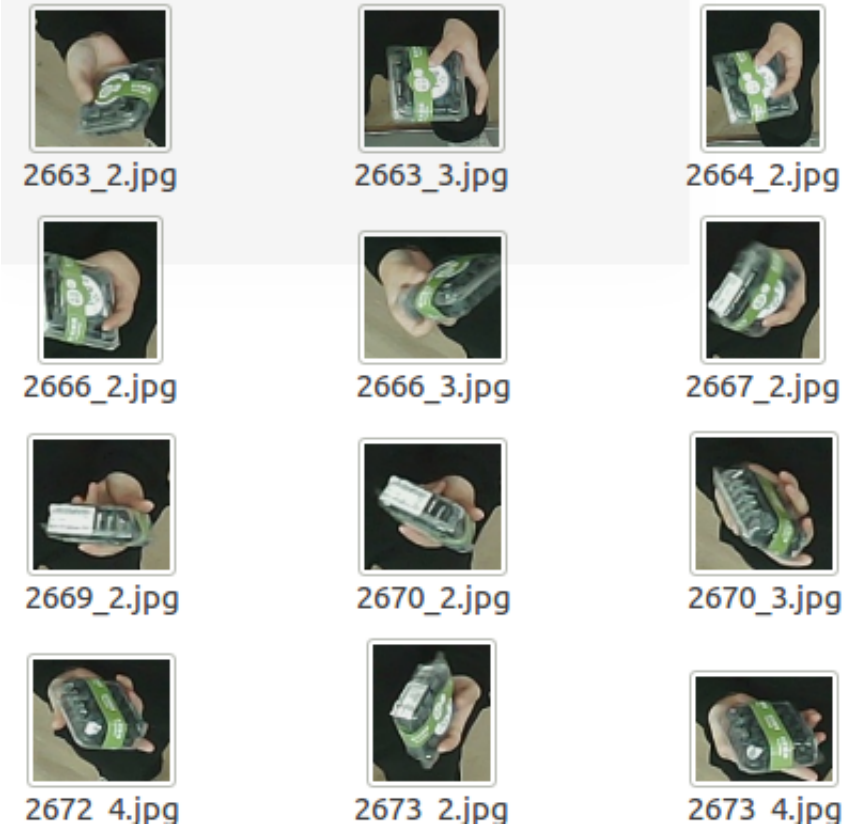
training的话就是为了看原来训练的是什么样 是不是有没有挑出去的负样本 点开看 就是这个东西！

所以是我们没把这个挑进去：

真的错误的蛮少：



训练的这样：



看来黑的特征太多了 而且可能是resnet18的局限

下周要做的：

1. 装相机，单独的识别

看能不能进行单独的输出（不依靠barcode 输出一个barcode）与正确的相比

2. 模型改进

1）在原本的yolo检测出来的手上 加简单的跟踪 输出每个手的set再进行识别 （混在一起很容易把另外一只手的东西当成要识别的东西了）当然 如果只有一只手在里面就不会有这样的可能0-0

2） 更改模型res50