膜色缺陷复盘:[2021.11.25]

* rgb颜色提取

1. rgb成像稳定性, 可能受镜片拍摄角度, 料盘底材, 相机成像模式等影响. 前期花了一段时间制定稳定的rgb取色方案: 吸光底材, 进口相机, 稳定的成像环境等.

2. 采用moyu哥给的get\_color.py脚本, 一张张过图像.

moyu\_ge建议: 绿膜数据拉高green\_lower就好, green\_lower值有一段区域变化但roi基本不变, 选这段区间的中间某处值就行了.

3. 后续考虑自动化提取rgb颜色不加入人工干预, 可能要上dl的分割来提取roi内颜色.

* rgb+ -> lab建模

其实是 rgb+ -> xyz -> lab, xyz -> lab 是一个固定参数值的矩阵乘法.

**1.** 选用的ML模型: xgboost

**2.** 精度度量方法: 10次交叉验证

**3.** 调优模型精度的过程:

1. 直接223条训练数据, 用之前的, 见过很多绿膜数据的"祖传"超参跑一遍, 精度0.88

2. 做训练数据清洗:

1. 绿膜数据出现g<b情况, 剔除这4条

2. LAB超出客户提供的正常范围, 剔除这11条 (集中出现在15 16 文件夹)

3. 15 16 文件夹数据, 相机成像模式与1-14不同, 剔除这39条

做完以上清洗后, 187条训练数据上的精度: 0.91

1. Moyu哥那边帮忙再check了遍rgb提取的config, 消除roi区域的”毛边”. 优化rgb值(这个点也很重要.)

4. 在"祖传"超参的基础上, 开启超参搜索(另外加了点人工调lr(很有用), 之类的..), 精度到0.936

5. 加入镜片roi区域的面积信息, 精度到94.2, 简单手动调了下参, 精度到94.6

* 一些快速实验和思考

1. 根据gamma变换后的gamma\_g或gamma\_b值, 设定阈值划分训练数据[这是之前蓝膜数据上的有效规律]. 根据gamma\_b/gamma\_g值划分数据也尝试过, 收益不大.

2. 拼接rgb值, 另外加g/b, g/r, r/b等信息, 使用k\_means做训练数据二分聚类, 分别调优两部分模型的精度. 数据被拆分为118和69, 精度分别到91+ 和93+. 整体精度没有显著提升.

1. 根据样本的lab值 **通过/不通过** 属性, 划分样本. 两部分数据都可直接到94+精度. 对于测试集, 先用两个模型各自预测lab值, 判断样本是否lab通过, 然后走专用模型预测样本的lab值. 后续验证此思路失败, 部分临近通过不通过的样本, 两个模型对lab值的预测, **不能反映样本真实的通过与否情况**. [通过/不通过属性划分样本在蓝膜数据上也一样表现较好，训练数据88张仅一张通过。直接训模型可到92.8精度].. 可见保持数据”不混杂”对精度提升很有作用. 遗憾这个通过否信息无法直接在rgb上体现..

**脑洞一下**, 当数据多了起来, 正异常LAB数据比例较均衡了. 可先训一个LAB正异常二分类模型, 再对此模型判定为是LAB正常的样本，做LAB值精准预测.~

4. 可视化各炉数据的rgb值分布, 发现各炉间颜色分布有明显边界. 这也是很重要的**数据特征信息**, 反应的问题是: 对于来自，训练集完全没见过的炉，的测试数据, 模型可能无法准确预测lab值. 因为各炉间数据“自墙分明”, 由于不同炉间生产环境的影响或不同炉生产的镜片特性不同, 导致不同炉间的”规律”较难迁移...

5. **要充分模拟客户设计的实验**。

对于蔡司留下的70条测试数据, 观察到: 11 12炉被完全保留测试，即模型完全”没见过”这两炉数据. 故在给客户反馈测试集结果前，可考虑**在训练集上设计类似的实验**, 完全保留某炉数据测试，其余炉数据训练，看看模型的预测能力. 更精细的, 寻找训练集中相似的几炉数据，选择其中一炉测试，其他剩余炉训练模型，看看模型的预测能力。

在本次客户验证前做了此模拟实验，确实发现模型对没见过的炉数据，lab预测能力较差. 后面在客户的测试集上，也出现了同样的结果，好在没太严重[11炉数据很特别, 错了8/20.. 12错了2/20]

* 模型调优总结

1. 清洗得到干净的训练数据很重要

2. xgboost模型, 精细调参很重要, 所以一定要上交叉验证, 防止过拟合.

3. 除以上, 就是针对项目特点的细节: 数据理解, 划分.

1. 模型输入端x, 对其分析, 划分, 扩充特征维度

1. 拆分出"相似"的数据簇, 在每一份稳定的簇内调优模型.

2. 预测lab值的上游任务: 二维图像 -> 特定长度的特征向量, 容易忽略掉除rgb之外的信息. 可以再添加一些roi形态, 面积等特征, 增加输入x的特征丰富性. 这一点需要结合客户给到的信息理解: roi区域的大小形状, 反应样本的度数闪光等信息. 度数闪光等因素可能影响镜片的光学特性[这里可能是我们关心的lab值]

[这一点在客户揭晓测试数据lab后发现, 虽然在训练集上精度调优1个点, 但在测试集上没有”成功”迁移... ] 所以**一定要注意模型的泛化能力**，但这和**充分拆解问题和挖掘数据信息**, 并不冲突...

* 后续训练模型模式

1. 前期验证/数据量较少场景:

1. 清洗数据, LAB异常的剔除, 蓝膜样本(b<g)的剔除, 绿膜样本则(g<b)的剔除. 训一个数据干净的模型, 精细调优各种超参, 精度尽量高.

2. 基于以上超参, 把上一步中剔除掉的(少量bg关系不符合膜系)样本加入重新训一遍模型, 再调整优化下超参之类的.

2. . 考虑后续online使用场景, 数据类型丰富充足

1. 可先做一个LAB值正常/异常二分类, 模型判定为异常的样本直接在此环节报出，模型判定为正常的样本则继续做LAB值精准预测.

* 最后

ML相关的**"数据"问题**, 很重要的部分在于针对特定任务 **理解,拆分 数据**. 一定牢记充分挖掘数据相关的信息, 为了精度,数据需要拆就拆, 然后再针对性的依次精准建模.

附:

1209 data 调优记录

1210:

1. train 231, 其中LAB范围异常 35. test 69. green data, g < b:0 [train: 196, test 69]

2. 用"祖传"超参直接跑, 精度: 91.3

3. 简单调了下2个params的lr, 精度到0.9163

4. 对param0开一个超参搜索, 搜一下lr值: 0.9188

5. 更新了下dir3 11的rgb config, 消除roi的毛边现象, 再加上调lr参: 0.9260. [开一个lr超参搜索, lr调整很有用.]

1213:

1. 优先做好LAB值正常范围内的样本精度(92.6现在), 客户留下的test种lab异常范围也不至于太多.

for ycy\_ in [1,11,22,33,44,55,66,77,88,99]:

res\_txt = open(r'./ycy.txt', 'w')

seeds = [i\*ycy\_ for i in range(10)] seed也可以做个最佳分布选择, 得到最佳acc: 92.7

2. 后续日常生产LAB异常镜片占比也不多的吧, 这类异常数据可持续收集, 数据到位后可训二分类模型, 第一步就做出异常报出.

3. 分析下0.927精度上高频出错的那几条样本. ('13\_11', 5), ('7\_2', 4), ('15\_3', 4)

1. rgb看着倒是正常范围, 所以感觉模型还有点欠拟合?

2. 分析出错的样本 l错多少? a错多少 b错多少.

4. 高频出错的13\_3, 针对它又重新回去调rgb颜色.: python get\_color.py "rgb" "train" "D:\work\project\卡尔蔡司膜色缺陷\1209\data\13\3.bmp" "D:\work\project\卡尔蔡司膜色缺陷\1209\data\13.json". 得到acc: 93.11

1213下午

1. 可视化下集中出错的那几条样本, 和蔡司留下的测试数据是否rgb相似 [貌似还好, 不太相似吧..]

2. 随机选几条数据测试下, 看看全量数据训的模型predict的test\_lab好, 还是多次seed1/4训的模型的test\_lab好.. [每次seed的结果感觉不是很稳定, 和mean的diff超过0.5的比较多..]

3. 要不先出蓝色的结果, 试试水~ [没弄, 绿色的交了, 现在来训蓝膜的模型吧.]

# 1215

1. 吴工给了绿膜结果, 分析下错误样本. 蓝膜对应调整下, 可以下午出一版结果到微信群里.

1. 总工错了27个, 其中LAB范围异常的有5个

2. 用seed的方式predict+mean, 输出结果看看. 错26个, 其中lab范围异常的错了: 5个。 所以结论是, 结果差不多~

3. 准备周四上午和蔡司会议对齐的材料. [done.]

1. rgb lab值 一起输出一份. [done]

2. 出一下蓝膜的test\_lab结果.[done.]

4. 总结一下材料发现然后需要跟蔡司对齐的问题点:

1. 本次采样数据和1118采样数据在rgb值分布上有明显差异, gb值均偏低

2. 测试数据lab预测误差超过0.6的样本, 存在明显的g/b比值异于训练数据的特征.

3. 测试数据中存在rgb lab不单调的情况. gb值接近, r值大不对应a值小.