基础算法科普

主编： 叶赛尔

献给无算法背景、无技术背景的广大人民群众...

以下是大致文档目录：

一.算法介绍：

1.公司算法组和产品组是怎么合作的？

算法组输出的一般是“模型”，也就是一个相对底层的lib库，实现相对底层的人脸识别功能，比如说，提供一个一张人脸比一张人脸的分值；

产品组则是基于算法组输出的模型，做上层实现，形成完整的产品，比如基于人脸1:1比对实现faceID这样的核身服务。

2.人脸识别的大致原理是什么？

如果去问RD这个问题，他们会告诉你 - “玄学”...

我站在我微薄的知识层面，大致科普一下这个流程：

- 检测 Detect/Track： 首先要从图片中找到人脸；
- 关键点 Landmark：然后人脸中找到关键点；
- 关键点处理：这里会有各种策略，比如最好理解的转正啥的；
- 属性分析Attribute：判断脸的Pose/Blur等是否合格；
- 抽特征值Extract：将人脸的关键点信息转换成一个N维向量；
- 比对Compare：比较两个特征值的距离，得出相似度。
- 活体Liveness：和识别并行的去进行活体检测，比如FMP；

一般我们划分算法为4个大组：

- 检测：把人脸检测出来的算法，Detect/Track/Landmark等；
- 属性：判断人脸属性的算法那，Pose/Blur等；
- 识别：1:1比对和1:N检索的算法，Extract/compare等；
- 活体：进行活体检测的算法，FMP等；

3.能大概解释一下深度学习是啥吗？

这块... 往深了说就不是科普贴了，给一个相对通俗的描述吧。

深度学习相当于建立了一个网络，在给定某些输入的时候就会给一些输出；这个网络有很多很多层，每一层都有一些运算规则，研究院做的事情就是制定这里头的运算规则；只是这个运算规则不是靠研究院小神童们自己精致的设计出来的，而是通过不停的喂数据，让这个网络自己“学习”出来的。我们可以把一个深度学习的神经网络视为一个智力有限的小孩子，人类通过不停的教他“这两个人应该看成一个人”和“这两个人应该不是一个人”，当这个小孩见识了大量大量数据之后，再看到两个人，就能判断是不是同一个了。

这个过程中，数据的输入极其重要，数据量要大、场景要吻合、质量要高，同时还需要人工标注，所以大家都会觉得深度学习的未来完全就是数据，因为基础系统的搭建到一定阶段，未来很可能谁有数据就是谁的天下。这也是为啥聊AI的时候大家如此在乎公有云、在乎数据回流，一方面是业务价值，另一方面自然也是数据训练的价值了。

当然这个过程中RD也极其重要是肯定的，同样一个小孩，学习快不快，老师的能力和教材的质量同样重要，缺一不可。

这里举一个老梗作为例子说，曾经有人把大量的哑铃数据放入到训练，训练出来的算法，很奇葩，只能识别“带手的哑铃”，光秃秃的一个哑铃是识别不出来的... 因为训练数据里面全是带手的...

这样的case很多很常见，比如活体组FMP的时候，拿faceID上用的版本到Koala上就有问题，因为faceID用户是拿着手机的，往往有手，但Koala场景经常不是...

4.数据集是什么样的东西？

其实想复杂了，真的就是一堆数据... 10w张Koala底库，1000w次faceID比对，1000断活体视频，都是。

不过可以科普一个这样的概念 - 数据集分训练集+测试集：

- 训练集：RD用于喂给深度学习神经网络的数据，往往要求量超级大，然后数据质量高；
- 测试集：产品组用于验收RD算法时候的集合，看算法实际表现是否符合预期。

然后有一个RD的节操：测试集不可加入训练集，不然的话测试妥妥过，但是实际产品效果并不一定好。

5.benchmark是什么？

简单讲就是算法的指标，作为RD的奋斗目标，和PD的验收标准。

比如对于1:1来说，我们往往会说的是，实现 “百万一误识率下，85%的通过”，这就是一个benchmark。

benchmark经常会被提起，是因为算法往往需要针对场景、针对硬件性能做各种平衡，benchmark不是一成不变的。

通常情况下，一个新产品的开发、或者一次大的迭代，产品组都需要制定becnhmark，作为目标需求提给RD，RD去实现、交付，PD验收。

这里也顺道说明一下，benchmark可以有很多的定义方式，RD的bm往往是算法级的，比如“RK3288上跑一次检测100ms”，但是产品级别的验收标准往往不是算法级的，比如“人进门过程总共800ms”。

这里的gap本质上是“一套算法使用策略”，是由PD+RD来一起制定的，举例说，如果技术实现上是：

“一共有4个核一个给用户GUI一个持续做track一个活体一个识别并行且不必考虑云端传输也大致忽略首帧以前的检测这样的话，检测75ms质量判断15ms活体180ms单核识别160ms单核大概能够实现800ms”，

这样子的话，后面的数值就可以作为benchmark提给每个组了。

当然实战不会这么粗暴的，凭空让而进在保持精度的情况下把识别从160提升到80，还不如杀了他...
小强也会跟你说FMP要单核单帧双千一就是天方夜谭... 这是靠one-team协调吧，给出最终有效解。

6.标注是做什么的？

标注就是对于一个数据的数据，进行人工标注，比如在一张人脸上标出关键点（为了孩子学会一样去找点），或者把两张一样的人脸放在一起（为了孩子学会这样两个人未来得算一个人）等。

这个过程是在一个叫Label++的平台上完成的，这是一个人力众包平台，标注任务在平台上发布，然后大量的标注人员会进行任务式标注，也会有人进行检查和验收之类的，总之，最终会输出一堆源数据加上一堆标注信息。

这是一个浩大的工程，但是经过一代代人的努力（呵呵，真是几波人过去了），目前已经是一个相对完善的体系了，包括网站本身的功能、线下人力调配机制、多底办公协作方式等等，整个Label++体系目前应该已经有300人的规模了，地跨多个城市。

二.人脸比对：

1.能结合“刷脸进门”这个场景说明一下前面提到的算法流程吗？

好吧。这样，当你出现在了公司门口的时候，摄像头捕捉到了你，给到算法的是一帧帧含有你人脸的图片，然后这个流程就开始了。

首先算法检测到了你的人脸，然后把各种关键点标出来，处理一下之后抽成特征值；

然后公司的N个人都作为一个底库预存在系统里，算法就把你的特征值和这N个人都比较了一次分数，把最高分的人认为是识别到的人，进行开门操作。

2.那大家说的通过率、误识率什么都是什么啊？是进不了门的人吗？

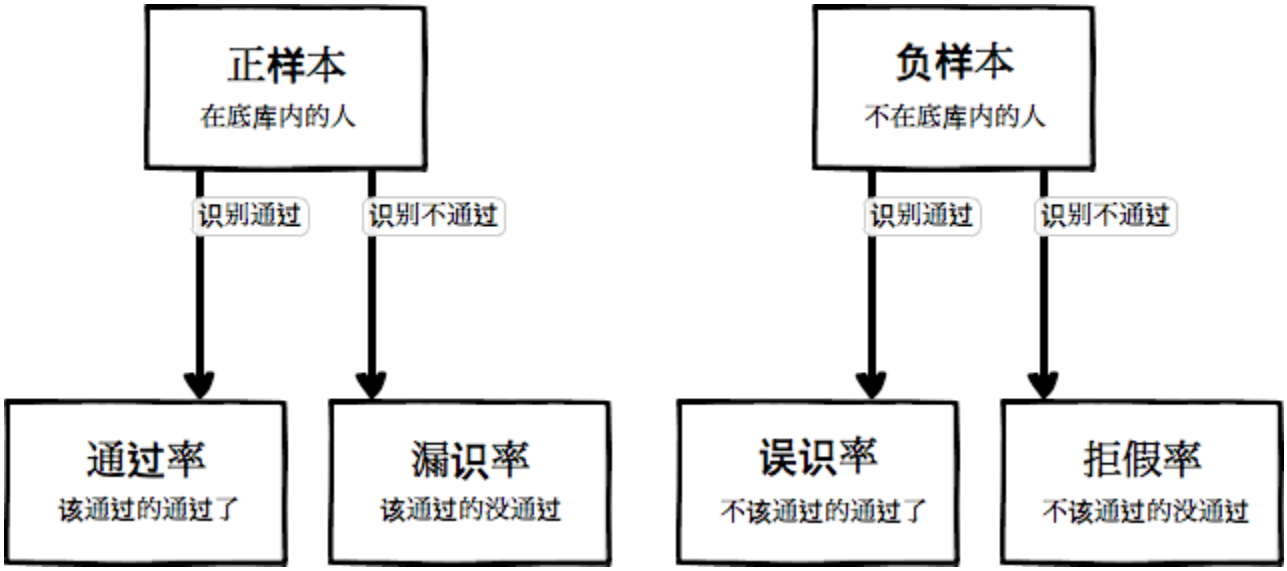
首先说，目前公司算法定一下的性能指标，绝大部分都是基于“单帧比对”的，而不是基于“人”的，一个人是可以发生很多帧比对的。

其次，所谓通过率、误识率什么的并没有一个严格意义的标准定义，出于不同的需要，大家可以自由制定，很多时候客户就要这么定义，你也没办法...

再次，但是通常RD会给出一个算法组本身的定义，如果用户的定义与之不同，我们就分开应对...

3.别废话了，就说RD一般怎么定义的吧！

好，那就给一下门禁场景下的定义吧，上图：



这就是目前FacePass产品定一下的几个率吧：

- 通过率 = 正样本通过人数 / 正样本总数；
- 漏识率 = 正样本未通过人数 / 正样本总数；
- 通过率 + 误识率 = 1；

- 误识率 = 负样本通过人数 / 负样本总数；
- 拒假率 = 负样本未通过人数 / 负样本总数；
- 误识率 + 拒假率 = 1；

由于有“加起来等于1”这个概念，我们一般不提4个率，而是只提：

- 通过率：衡量目标用户通过比例，直观理解为“公司员工顺利通过门禁的概率”，越高越好，低了会出现用户进不了们的投诉；
- 误识率：衡量非目标用户通过比例，直观理解为“外部人员通过门禁机的概率”，越低越好，高了就是安全问题了。

不同场景对于通过率和误识率的要求是不同的，如果你能在3s钟内想明白“为什么门禁机追求误识率、而考勤机追求通过率”，就说明你对算法和场景已经融会贯通了。

4.恩，感谢科普，可是...我开始有点迷糊什么叫“通过”了，不是识别出来了就算是通过了吗？

门禁机场景下，“通过”定义为，底库比对的最高分top1高于通行阈值，反之则为不通过。

可以这么理解，1个人脸去N个特征值底库中比对，无论如何都会有一个最高分，我们要看最高分是不是“够高”，够高就视为准入，不够高就当陌生人不准入。

5.能再科普一下“阈值”的概念么？

首先，除了曹志敏以外，希望所有人都不要念“阀值”，这个会给专业人士带来巨大的困扰...

其次，在人员通行场景，阈值是一个分值，用于判断一个人脸识别比对的结果“是否应该放行”，可以理解为这是一个产品层的数值，而不是算法层的。

什么意思呢？

算法说，和你最像的那张人脸，比对分值是81.2；而产品这边认为说，这个场景下，过80分的人我才能让他进门，阈值80，81.2>80，所以你就进门了。

6.那看着阈值很霸道了，比如上面，如果阈值设计为82，就直接进不来了？那一般有什么设计原则吗？

阈值的确很霸道，举例说，如果阈值设计为0，那么所有比对分都大于0，也就是说，不管任何人都能进门；相反的，如果阈值是100，那么所有人都进不了门...

实际操作中当然不会这么极端，比如Koala v2的基本阈值是78，FacePass SDK可能是81之类的。

要讲清楚阈值设置原则，对于无算法背景的人来说，就需要搬个小板凳来慢慢说明了...

7.别废话，说吧~

好，整体科普了一下阈值和通过率、误识率的关系。

首先，（不管是正样本还是负样本）阈值越低，人越容易通过；阈值越高，人越不容易通过；这应该都能理解。

其次，对应的，阈值越低，通过率越高，误识率越高；阈值越高，通过率越低、误识率也越低，两者一个正样本一个负样本，同升同降。

再次，通过率越高越好，误识率越低越好，这是场景诉求，应该也好理解。

于是问题就来了，你会发现说，阈值上下调节，这俩值都是同升同降的，如何同时实现高通过率和低误识率呢？

回答是：通过选择一个“精妙”的阈值，使得两个值综合最优。

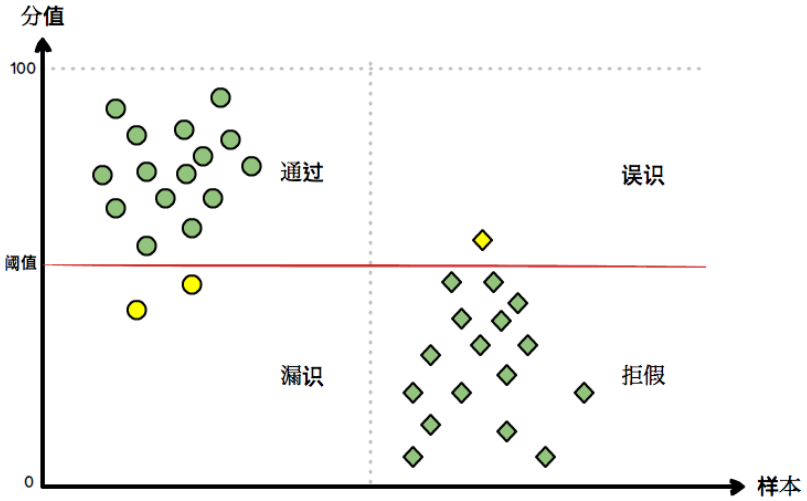
这里产品组是机械调校师，在一辆赛车已经给定的情况下，通过各种策略，让算法的表现达到当前场景下的最优；

但是算法本身决定了这个调校的上限，一辆二手奥拓怎么调校也不会比兰博基尼跑得快。

所以这件事情上，需要RD+PD的综合实力才能达到最终完美用户体验的。

8.ok... 我大概明白了，可是... 有更直观的解释吗？

有，那还是上图吧：



圆圈代表正样本的识别分，菱形表示负样本的识别分，红线表示阈值，阈值以上的表示通过，以下的表示不通过。

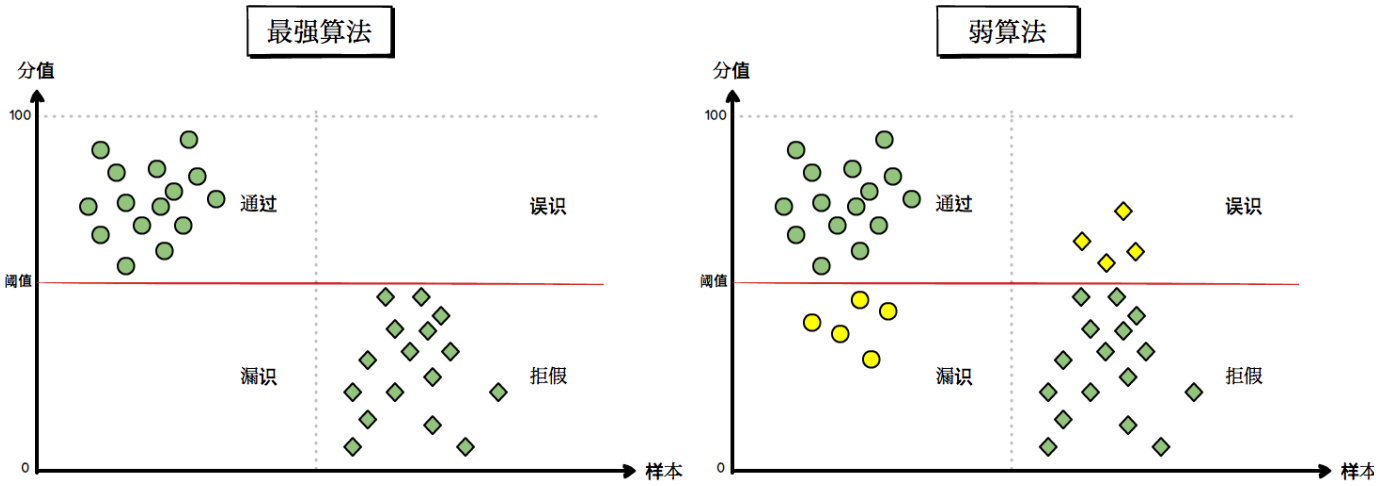
识别分对于给定模型版本来说是不可调节的，但是阈值是可以调节的。

阈值可以上下浮动，最终目标，是“让尽可能多的圆圈停在红线之上，同时让尽可能多的菱形停在红线以下”：

- 红线在0的时候，所有样本都通过了，所有正样本均通过，通过率100%，所有负样本也均通过，误识率100%；
- 红线往100上移的时候，开始吞噬一部分的菱形和圆圈，我们默认一个靠谱的算法，对负样本的识别分整体是比正样本低的，所以这个上移过程会优先吞噬掉大量的负样本，表现为“很不像的人被拒绝通过了”；随着阈值越来越高，开始接触到一些模棱两可的人，有点像也有点不像...
  这是考验算法能力的关键，越能区分开越厉害，但是事实上再厉害（即使厉害到旷视这样的），往往也无法完全区分开来，所以如图所示，红线所在位置，其实就“错误的”漏掉了两个正样本、和误识了一个负样本。
- 再往上移的时候，就开始更多的吞噬掉正样本了，如果到达100阈值，那么所有人都视为不通过，统统拒之门外。

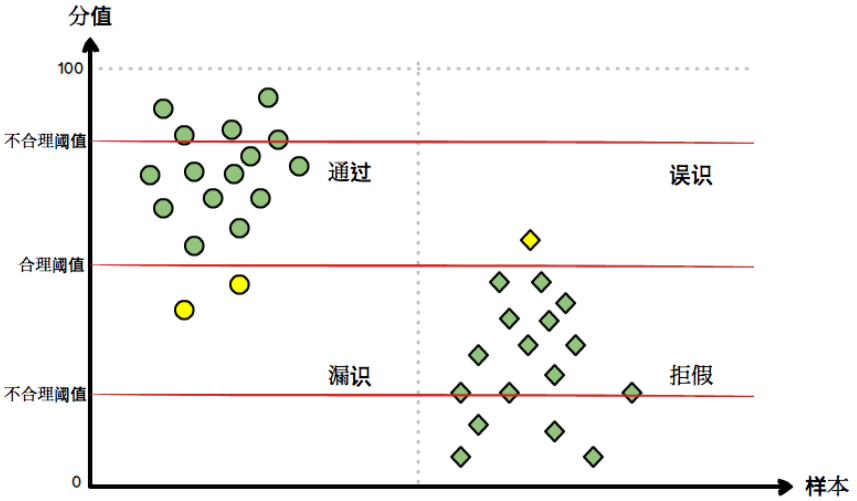
9.这个我大致明白了，那么之前说的好算法和坏算法（也就是兰博基尼和二手奥拓）的区别什么？

一张图说明，聪明的你，不需要我多解释了吧！



10.ok... 那么对应的，产品的算法调校（好的机械师和烂的机械师）差别又是什么呢？

同样一张图说明：



不过，还是需要说明一下，“脱离场景聊阈值”都是虚妄的，站在数学角度上说，我们认为阈值设置的三个条件：

a.正负样本比例；

b.正样本通过得分；

c.负样本误识减分；

有这三个值的时候，加上算法本身的阈值曲线，阈值其实是可以反向计算出来的 - 这个事情广泛发生在各类算法pk中。

举一个实际案例，广东白云机场安检前台，用的1:1核身比对，正样本是正常安检用户，负样本是持他人护照的客户（攻击！）。

作为一个PM，会这么思考问题：

a.这个场景下，正样本远远多于负样本（攻击极少），比如十万比一；

b.正样本通过得分正常，真错一个这不还有机场工作人员帮忙核对吗，比如 +1；

c.负样本误识扣分严重，因为这会让丁副市长逍遥法外，比如 -500；

所以最终的得分计算方式就是：

V = 通过率 * 1 * 1 - 误识率 * 500 * 十万分之一 （公式A）

一般算法会给出的曲线是这样的：

68阈值下，通过率98%，误识率千一；

78阈值下，通过率92%，误识率万一；

88阈值下，通过率85%，误识率十万一；

代入公式A中得出最大的V即可（当然实战是连续曲线，不是这样的离散数值）。

11.之前有提到1:1和1:N这两个概念，能否专门的说明一下？

ok，好。

1:1指的是对两张图进行特征值相似度比对，给出相似度；

1:N指的是在N张图中检索和目标人脸最像的特征值。

如果站在纯粹算法的角度，两者是相通的，N次1:1可以实现1:N，但是，同时也有本质的不同，这个再下一个问题专门阐述，有点复杂，得引入非人类语言。

这里先回到人类语言解释产品级概念。

1:1目前最常见的应用，是身份核验，简单讲，就是目标用户的人脸、和系统人脸，到底是不是同一个人。

目标用户人脸往往是摄像头拍摄的，系统人脸则是比如用户身份证读出来的、或者去公安系统调用的、或者VIP会员系统存储的。

所以常见的是一体式的人证前台机、或者机场人证核身系统，大家可能都见到过。

大部分这样的场景都有人值守/或者有时候人工校验，且“判断两个人相似度”这件事情算法来讲的确不是那么难。

所以说，这块技术总体上说相对成熟的，目前在金融（比如银行、互联网金融等）、出行（机场、火车站等）、商业（酒店、访客等）也都开始广泛应用了。

1:N就复杂一点，是从N个中找出最像目标人脸的一张人脸。

从技术上将不仅要保证每个1:1品质（比对分靠谱），也需要保证“我比我”要准于“别人比我”，以及“陌生人比谁都不像”，这块一会儿画图说明。

同时，这块场景上的要求往往也比1:1来的困难，比如典型的安防场景，目标往往是30w的全国在逃犯底库，30w人啊，里面怎么都会有很多很想的人啊...

再比如门禁场景，一个完美的门禁，是“无误无漏”的，要求通过率极高、误识率极低 -
刚才我们已经描述过，这俩同升同降的，不能像人证机1:1那样可以损失通过率。

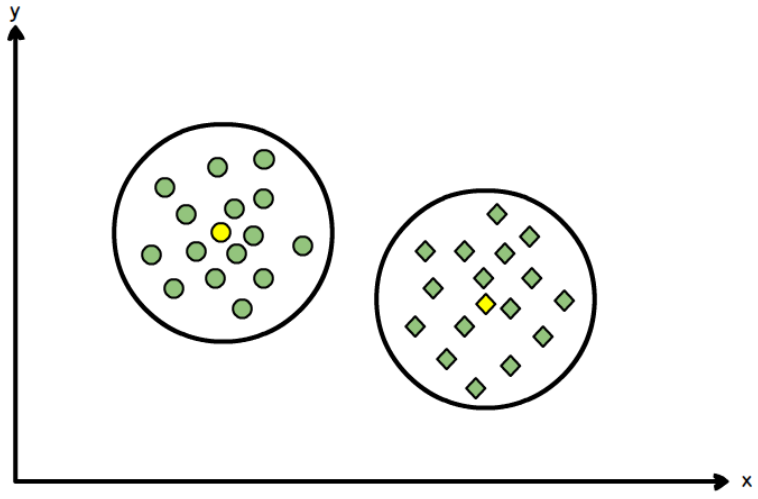
大家幻想中的FBI天网、各种未来科技啥的，也都是1:N，毕竟核验是第一步，最终还是要检索的。

这块全世界的科学家、教授、小神童、工程师、产品经理，都在试图一起提升到商业上畅行无阻的水平，这几年AI也因之突飞猛进，但是距离未来科技还是距离很远的...

目前比较典型的场景，在公司层面来讲是安防人像卡口（黑名单抓逃犯）、商业线人员通行、零售二次到店和会员体系、手机相册聚类等。

12.好，可以用图解来说明1:1和1:N了。

我们用一个更加深刻的模型来描述人脸识别的定义，之前的数值型阈值是一维的，很难来描述人跟人的关系，这里引入二维的：



在这个坐标体系里，任何一个人脸特征值都对应于二维平面上第一个坐标点（实际上是几百维，但我是三维生物，只能凑合画在二维平面上了）。

然后任意两个点的相似度，在这个平面上就等于两个点之间的距离 - 这个也好理解。

那个黄色的点是什么呢？通常表示“质量比较好的人脸”，可以作为一个标准，计算其他点和他的距离，比如身份证高清大图。

然后绿色的点呢？大致可以理解为不同的抓拍人脸，通常默认抓拍质量越高（越清晰、越正脸等），就离黄色点越近。

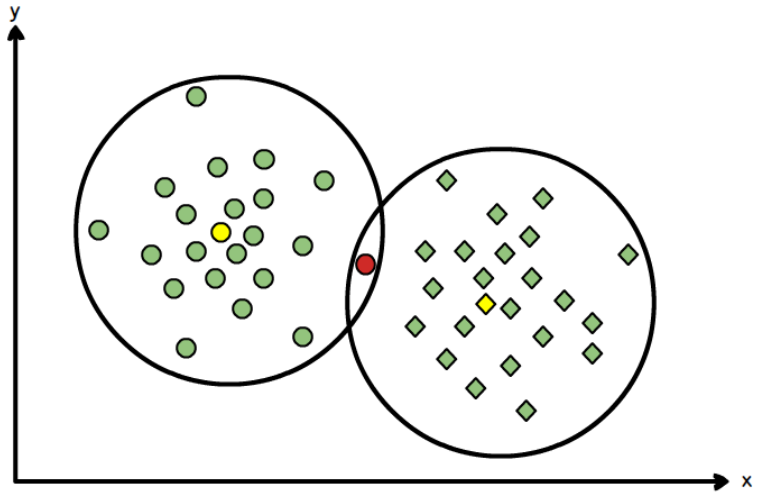
那个圆圈又是什么？我说阈值你信吗...
我们希望有一个圆圈，把那些比较接近的抓拍都圈起来，所有落在圈内的都是本人、落在圈外的都是他人。

合格和画一条横线做阈值、高于阈值都是本人低于阈值都不是本人，本质上是一个逻辑。

同样的，我们发现画一个足够足够大的圈，总能把自己的抓拍点都圈进来，但是免不了把别人（比如菱形点）也圈进来，这个本质上也是一样的。

所以，上图表示的，就是两个用户A和B，以及他们各自的抓拍图，以及各自用一个圈阈值把他们分了类。

好，下面进入一个复杂一点的情况：



实际上，很多时候算法的点阵分布很难那么完美，比如上图，为了让同一个人的所有抓拍都进入阈值圈，这个圈就得比较大。

这个时候，可能两个不同的人（圆形和菱形）的阈值圈就重合了：这代表什么呢？

代表那个红色的圆圈，同时满足圆圈先生和菱形先生的阈值要求，也就是说：

圆形先生红色点对应的那次抓拍，如果拿着菱形先生的身份证去做1:1核验，是能过人脸核验的！！！

这就是一次负样本的通过，也就是大家日常说的攻击成功，也就是一个误识的case。

类似于之前一维模型，我们也看一下兰博基尼和二手奥拓的差别、也看一下机械师的工作是什么。

算法组角度说：

一个优秀的算法就是“让绿色点尽可能的围绕在黄色点周围”，以至于“阈值圈尽可能小到没有重合”，当然，这都是理想状态。

产品组角度说：

既然理想状态达不到，产品就要在阈值层面平衡漏识和误识。圈越大，越能保证自己的抓拍被圈进去，漏识率月底；但是也越容易和别人的圈重合，容易出现刚才那种红点，也就是误识。

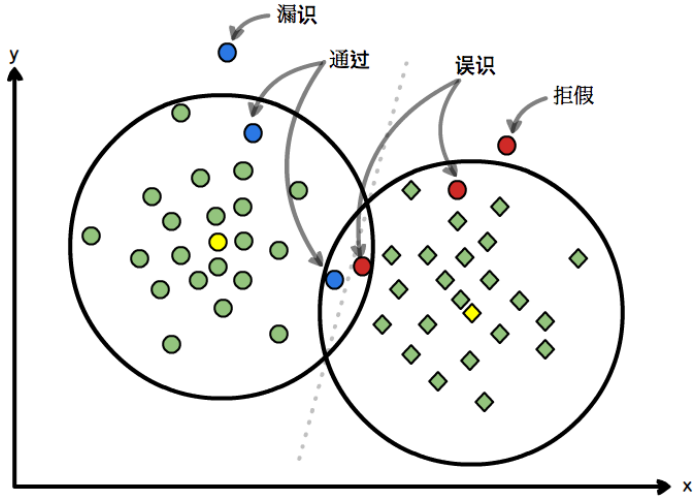
所以产品对这一点的帮助，就是平衡好误识和漏识，给出最合理的阈值，画出最合理的圈。

然后可以开始说1:N了。

1:N有top1的概念，也就是找到“最像的一个人”，翻译成这个坐标系下的语言，就是距离最近的一个黄色点。

所以，所谓的1:N，可以简单描述为，对于任何一个抓拍点，找到距离最近的黄色点，如果落在阈值内，就算通过（如果的确是本人，就是真人通过；如果不是本人，就是他人误识）；如果没有落在阈值内（而是落在各个圆圈的空隙处），则表示不过阈值，不予通过（如果的确是本人，就是漏识；如果不是本人，就是拒假）。

我们画个图说明一下吧：



如上图所示，给圆圈先生和菱形先生的点阵+阈值圈，然后针对圆圈先生新的抓拍人脸，我们会有如下几种可能：

- 蓝色：表示“这个抓拍更像本人”，也就是说，离黄色圆圈要比黄色菱形来的近；
- 红色：表示“抓个抓拍更像其他人”，也就是说，离黄色圆圈要比黄色菱形来的远；

*请额外注意两个圆圈交集部分，对于1:1来说，只发生一次比对，落在其他人的圈里就误识了，对1:N来说，进行了N次比对，本质是在看最小距离。

所以：

- 蓝色圈外 = 漏识：识别成自己，正确，但是不过阈值，不让通过，坏结果；
- 蓝色圈内 = 通过：识别称自己，正确，且过了阈值，让通过，好结果；
- 红色圈内 = 误识：识别成别人，错误，且过了阈值，让通过，坏结果；
- 红色圈外 = 拒假：识别成别人，错误，但是不过阈值，成功拒绝，好结果。

如果这一刻你很对这张图表达的意义一目了然了，那么恭喜，你对于xx率这块的理解已经可以毕业了。

最后补一下说，为什么1:N这么难呢？以及好的算法到底会好在哪里？

你是否理解，1:1本质上大约是N=2的情况？（不是N=1）

如果一个1:1很准确，意味着“进行识别的两个人的阈值圈没有重合”，也就是说，在保证不太漏识的前提下，不会出现“同时过两个人阈值的抓拍”。

同时，大家应该很容易理解N越大，识别准确度越低，映射到图上就是：

- 有越多的圆圈，一个新的抓拍点就越容易离“其他点”；
- 有越多的圆圈，阈值圈越难“协调”，因为你越容易overlap；
- 其他...

综上，1:1大致等于N=2，N越来越大自然越来越难，尤其是到N=20000的时候，这个就很难了。

关于识别率相关的讨论，到这里就告一段段落了...

三. 其他算法：

1. 继续？我想知道识别以外的哪些东西，检测、属性、活体之类的。先检测？

可以。

检测一般我们说分为detect和track：

- detect：在一张图片范围内，检测人脸，一般用于检测“新出现的人脸”；
- track：轻量级detect，往往约束检测范围（比如detect附近的范围），一般用于“人脸跟踪”；

这两者其实底层实现是一样的，但是部分输入、策略和参数不同（据我目前的认知），track会传入上一帧detect/track的信息（人脸框位置啊、内部参数等）。

简单讲可以这么理解：对于一个人，进入画面时候找到他，靠的是detect；然后他在视野范围内走来走去，跟踪住他靠的是track。

可不可以每帧都用detect呢？

理论上可以，但是有两个原因让我们选择track：

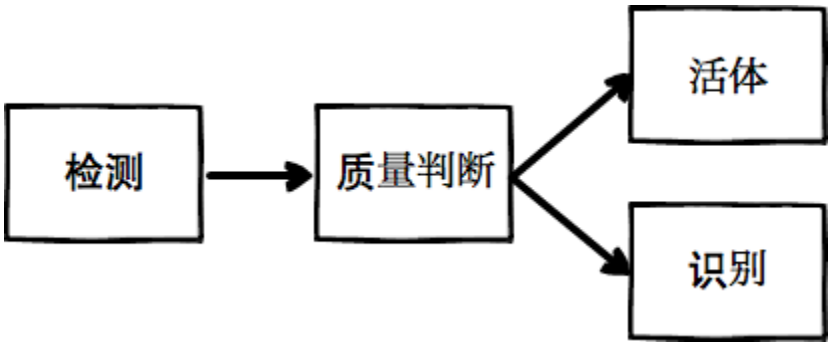
- 我们默认人不会飞速运动，所以逐帧相对连续，我们不只是要找到人，还需要利用连续性把人串在一起，作为一个track，赋予一个trackID。
- detect比track慢的多，以RK3288为例，detect70ms， track7ms，基本是10倍差距。

所以一种策略也是说，detect可以不必每帧都做，因为一般可以容忍“一个人新出现的延时”，但是tracker是必须每帧都做的，因为为了跟踪的好，两帧之间的人得尽可能的接近，所以理论上帧率月底越容易跟丢，极端一点，比如全志A20这样的板子上，默认一秒钟只有两帧，就可以直接放弃track策略，直接上传单帧图片使用。

2.能多解释一下为什么要存在一个人的track呢，每一帧都利用起来不好吗？

简单讲，是为了“让人脸检测的输出减少数量、提高质量”。

一个标准的流程是这样的：



检测是所有后面流程的输入，漏斗的第一层；质量判断是第二层；检测完成+过质量判断之后的人脸，就真的要被拿去做识别和活体了。

给一个RK3288上的性能参数：（单位毫秒）

算法类型	检测	质量判断	识别	活体
单次耗时	75	15	150	180

明显识别和活体是很占时间的，所以一般来说，我们希望给到识别/活体的是相对有效的人脸。

同时，门禁场景本身就是“以人为单位”的，如果track有效，那么每个track一帧就可以了...

至于说“提供质量”，是指说，一个track里面不见得每一帧都合适做识别，所以不能无脑送，要先进行质量判断才行。

所以个标准的逻辑是：track里头的每一帧都送质量判断，过了质量判断的第一帧送到后方去做活体和识别。

3.检出率、误检率这类的参数呢？和之前的识别相类比？

介于之前花费了太多口舌，这里就相对简单一点的说一下。

类型	定义
检出率	该检出的人脸，被正确检出的比例
误检率	不该被检出的人脸，被检出的比例

类比于之前的识别，这俩也是同升同降的。

而且，一般来说，误检率很难被定义，因为攻击方式（比如狗脸猫脸、像脸的树叶、应急灯之类的）是无穷无尽的，不知道分母该怎么定义。

通常就是找一个（或者采集一个）的数据集，量足够大，用来大致作为误识率的benchmark。

4.之前有提到过track中断率，这是个什么？

理想状态下，一个人进入摄像头范围，默认只有一个track，从进来开始跟，跟到出去为止。

但是理想状态很难达到：

- 人脸的一些表现可能无法达到track的需要（比如过度低头、转头，恰好被遮挡等）；
- 摄像头实际帧率不足，或者cpu等性能跟不上引发降帧等，最终会导致两帧之间间隔太长，以至于track不住；
- 算法无法完美；

所以，track有概率会中断的。

track中断有什么影响呢？

- 觉得不爽；
- UI上看，框会中断，不够精致；
- 交互上只能更复杂，比如门禁机开门，一个人当然只能开一次，如果断成俩track了：
  - 运气好都比对成了同一个人，比较UID一致，第二次就不开门了；
  - 运气不好两次比成了两个不同的人，就只能无脑的开两次们了，还是两个不同的人进门....
- 但是以上都不本质或者可以克服，最要命的原因是“无形中加大了端+云整体的负载”，对于系统来说，断掉的track就是多个人，自然也用更多资源来当多个人处理了。

一般我们定义track中断率为：

track中断率 = 断track导致的新增track / 总track数

直观理解为：多大比例的track是新增出来的。

比如9个人通过，有一个人track断成了2个，则中断率为 1/(9+1) = 10%，所有track中有10%是“断出来”的。

对于门禁机来说，这是一个很核心的参数，高了对云和端的性能都会造成不小的影响。

5.说一下属性？从pose开始？

pose，也叫3D-pose，简单讲就是人脸在3D世界的角度检测，门禁机场景主要用于判断角度是否过大、需要丢弃。

pose有三个维度：

- roll：人脸平面上围绕脸部中心的旋转；
- yaw：左右转头；
- pitch：上下点头；

换成数学语言，如果鼻子到耳朵是x轴、鼻子到下巴是y轴、垂直x-y的是z轴，那么：

roll是绕z周转，yaw是绕y轴，pitch是绕x周（这么说是让事情简单了还是复杂了....）

一般来说pose的评价标准，就是度数准不准，目前的水平是大约5度以内的误差。

再往下做就比较难了，因为比较难以标准也难以验收，人脸出现5°左右的差别，还是很难感知的。

最后说一点，不同场景对于pose的要求不一样，比如faceID基于手机，很多时候对仰角支持的好一点，安防架杆摄像头，需要算法支持更好的俯角等等。

6.好，说下一blur？

blur，blurness，模糊度，是目前属性中比较关键但是也比较难搞的一点。

blurness产生的原因很多，比如因为过快运动导致的、比如失焦造成的等，不过RD并没有给出更细的接口，而是统一的值。

blurness之所以难搞，主要是因为“很难定义标准”，100个人对1张图的模糊度的打分可能都不一样，同样，1个人对100张图的模糊度理解也不一样。

难以定义标准，糟糕的不是说出来的结果众说纷纭，而是“无法well-define的标准会难以标注”，导致blur很难有精准有效的数据集，训练和优化难度就大。

目前采取的标准方式是五分类、或者糙一点的二分类，简单讲就是把一张图片的模糊度分档（而不是打分），用于训练。

现阶段公司blur的水平感觉一般，实战中各个组都有反馈困难，但是实在也不知道怎么快速提升。

7.其他属性呢？

pose和blur是门禁机场景中最关键的俩，因为这俩对识别、活体的影响是最大的，其他属性第一轮其实是没有加的，他们包括：

- Illumation：光照，比如光照强度、过强过暗、阴阳脸等；
- Occlusion：遮挡，比如眼镜、墨镜、口罩、刘海等；
- Age：年龄，目前小孩和老人都是不准的，主要是中青年，但是年龄差距也会在10岁以内，不是那么准；

- Gender：性别，目前也一般，这条线没有怎么商用，感觉商用有差距；
- Others：其他的我就没有关注过了...

这里其实是分两类的：

- 一类是用于算法内部控制的，比如pose、blur、occ，在约束范围内的人脸才能送识别；
- 一类是用于提供产品API的，比如age、gender，在零售领域就有广泛的需求。

8.刚才提到的小孩老人不准，是为什么？

有主观和客观的两个原因：

- 客观上：小孩是因为脸部特征不明显（脸盲），且脸部变化大（自己不像自己）；老人是因为纹理太多，容易干扰；
- 主观上：由于算法不成熟，我们也就没有落地场景，也就没有数据，也就没有训练，也就没有优化，恶性循环啊....

9.外国人呢？

楼上的观点也可以用于外国人，这块怎么说呢，也是缺乏数据，实战表明，来哪个人种的数据，就能优化哪个人种，且目前并不区分模型。

但是人种的区分度其实还挺大的，比如东南亚、印度人和目前汉族都差异巨大，需要单独的数据集进行训练。

所以说，目前公司算法大致是：正常光照下、清晰、中青年、正脸、汉族人脸识别系统，当然扩大这个边界也不是不能用，但是精度和这个集合相比还是会下降的，任重而道远啊！

10.剩下是不是就是只有活体了？

是的，活体liveness，这是最近过去这1-2年异军突起的一个方向，原因是人脸识别落地过程中，安全性其实被广泛质疑，无论是faceID线上核身还是Koala门禁机白名单通行，每当一个人脸识别产品诞生的时候，就会有对应的黑产进行对应的“攻击”，试图破解，这是一个长远的工作，道高一尺魔高一丈，永远有黑产，也永远有算法优化，作为一个长久的方向来支持。

活体目前来说大致有如下几个发展阶段（并不是严格的时间序上的）：

- 配合式活体：要求用户配合式的做动作，算法检测对应的动作正确性，比如faceID活体；
- FMP活体：单摄像头，根据单张图片判定活体，比如手机解锁活体；
- 硬件活体：红外活体或者双摄，利用其它硬件特性来完成，比如iPhoneX活体；

【配合式活体】

大致上算上一代技术了，往往是要求用户给出“抬头”“眨眼”等配合动作，然后进行动作的连续性判断实现。由于黑产发达，所以用各种视频方案来攻击。

现阶段的唇语活体是一种升级，因为给出的文字量会更多，比如4位数字就是10000中可能；相比于一些标准动作（比如抬头摇头），会更难以攻破。

配合式除了安全性以外，一个很大的问题是，很多场景都是不可动作配合的，比如安防和门禁机场景，很多都是无感知的。

即使是一些有感知的场景，配合式活体的体验已相对冗长，除非像金融、账户之类的的“严肃”操作，很多日常使用多少比较奇怪。

【FMP活体】

则是现阶段公司的主要技术，最早是F-Mask-Panaroma三个字母组成，但是后来早就不是当时的定义了，现在在不停地重造词语赋予FMP含义...当然，这个不重要。

FMP异军突起是因为它代表着单RGB摄像头下的基础形态，适用于手机、pad、闸机、枪机等诸多形态，且能够和RGB人脸识别方便的互动。

所以过去一年里内，公司的活体精力大部分投入在FMP活体，在手机解锁、钉钉门禁等各种方向都有落地，进步明显。

但是FMP的天花板效应也相对明显，在没有额外设备输入的情况下，要靠单张图片搞定问题实在比较困难，目前单帧误识基本还在百一量级，复杂case（扣眼面具）有十一的，离默认千一的产品可用度差10倍。

大家始终感觉需要同步储备红外、双摄等硬件活体来支持更完整的活体体验。

【硬件活体】

这块也就是个泛称，实际上有很多的形态：

- 炫彩活体：通过往脸上打光并检测；
- 红外活体：通过一帧红外一帧RGB并比较差别；
- 双目活体：通过两帧RGB在一定角度下的差别判定；

之类的吧，挺多的，个人感觉这块公司尚未真正实现可商业化的产品，红外一度想落地，但是发现距离还很远。炫彩和双目我不太清楚，不多说。

有一种说法是，未来终究会硬件化的，这个更多是基于“单目摄像头（FMP）活体有天花板”这个假设提出的吧。

11.那活体的FP/FN啥的又是啥呢？

好，复习一下之前说的各种概念了，一分钟内梳理完：

正样本：活人，正常使用的用户，希望不要被判定成攻击；

负样本：死人，攻击用户，希望被判定成攻击；

False Positive：该是负样本，但是算法判定成正样本了；

False Negative：该是正样本，但是算法判定成负样本了。

同样的，FP和FN也是一个trade-off，虽然不见得一定FP高了FN就必须低（至少不像识别那么明显），但是算法都占资源，每个模型版本在性能不变的情况下，效果多少都会有些侧重。

（持续更新）

罗列一下部分名词定义，供大家查询：

RD	Research Develop	算法开发，也代指公司研究院/算法组
PD	Product Develop	产品开发，一般意义上的技术开发组
PM	Product Manager	产品经理，一般也是算法这边的需求方
Detect		人脸检测，从一张图中找到人脸
Track		动词，人脸跟踪，轻量级detect，在约束条件下快速跟踪检测人脸
		名词，同一人脸的连续帧统称为一个track
Attri	Attribute	属性，比如角度光照模糊遮挡之类的
Pose		角度，属性之一
Blur	Blurness	模糊，属性之一
Illum	Illumination	光照，属性之一
Occ	Occlusion	遮挡，属性之一
Age		年龄，属性之一
Gender		性别，属性之一
Liveness	Liveness	活体，检测识别的人脸是真人
FMP		单目摄像头活体技术，Liveness之一
FRR	False Rejected Rate	正样本被拒绝的比例
FAR	False Accepted Rate	负样本被通过的比例
FP	Fase Positive	负样本被通过的比例 = FAR
FN	Fase Negative	正样本被拒绝的比例 = FRR