

Z-AI1.4 版目标分类器建模

文档版本:1.0

最后修改时间:2024-2-9

修改日志

2024-2-16,第一个版本完结

目录

什么是目标分类器	4
目标分类器的推理	5
弱推理	5
中推理	5
强推理	6
目标分类器的数据规格	7
训练分类器的数据前置处理	7
ResNet-Metric	7
Large-ResNet- Metric	7
GDCNIC	7
GNIC.....	7
Z-Metric-V1.....	7
Z-Metric-V2.....	7
Prepare Rasterization 小图数据	8
拟合框抠图机制	9
生成 Prepare Rasterization 小图	9
通过手动画框来成 PR 人脸小图.....	9
通过人脸工具自动生成 PR.....	10
通过手画 4 顶点生成 PR.....	11
4 顶点的 SP 模型	12
目标分类器的反推机制	13
保存向量为什么使用 Learn 而不是 KDTree 这类二叉树结构	13
在良好的 Metric 模型基础上,Learn 的第二次建模	13
在 Model Builder 标注目标分类器	14
目标标注镜像	14
目标标注抖动	15
ZM2 的目标分类器.....	16
在图矩工具用大数据标注目标分类器	16
在图矩中堆砌目标分类器样本	16

图矩工具中的尺度系统	17
在 Model Builder 生成小图并导入图矩	17
在 Model Builder 生成带有抖动和标签分类样本库并导入图矩	18
训练目标分类器	19
测试目标分类器	20
简单测试	20
大数据模型测试	21
测试人脸模型	22
业务系统中的目标分类器	23

什么是目标分类器

分类器是机器学习和统计学中的一种程序计算模型,例如在 NLP 中单词可以是名词,动词,形容词等等,在 CV 中照片分类器可以对一类图片进行场景分类,例如森林,城市,街道,房间.而常用的人脸识别,是从图片中根据检测器的框框抠图,然后再识别这是谁.

分类器的本质是一种程序计算模型,Z-AI 体系的分类器有:场景分类器和目标分类器,这些分类器都是作为一种子程序,在不同的位置发生作用.

场景分类器通常会充当先头部队,先识别场景,例如判断夜晚或则白天,街道或则广场,对一整张照片做判断就是场景分类器,当程序知道工作场景这时候就可以做出分支流程了,例如是道路,就执行机动车检测器,是广场就执行人检测器.

目标分类器通常作为一种战场扫荡部队,直接产生最终的 AI 数据,这些 AI 数据大都会接口到业务层和数据库层.目标分类器的触发程序模型通常为检测器,例如人脸识别会先检测人脸,然后抠图交给目标分类器,另一种触发程序模式是坐标锚,这是一种固定锚,例如微信扫码的中央框,某些手机解锁需要你头像对准人头槽,某些监控摄像头需要画框,这些例子都是一种坐标锚,通常按固定时间抠图,例如每隔 0.5s 从坐标锚抠一次来识别,然后直接交给目标分类器,直接使用坐标锚可以绕开检测器,这样可以识别提速,另一方面,通用检测器建模对数据源,算力,标注,都有许多要求,建模难度偏高,因此直接使用坐标锚.

对于一些对于精度要求不高的场景目标分类器也是可以作为场景分类器使用的,例如把 1080p 或 4k 的场景图片,缩小成 150*150,然后做场景识别,多数时候还是建议使用专门的场景模型,因为目标分类器算法思路不是场景识别而是识别小图块.场景识别会有专用算法.

对于一些针对人类肢体和动作行为类的模型,底层会使用目标分类器,例如拉着皮箱行走,双手插兜行走,交警在指挥交通中所摆出的指挥姿势,这类行为通常使用视频作为数据源,从视频中连续性的挖掘从抬手到收手的一系列动作,然后,再序列化这些动作,最后跑业务时再从识别结果数据库抽取一连串的行为姿势识别结果统计处理,最终的结果会是某个人在某个时间做出了某些肢体行为动作.市面上有许多 AI 盒子,号称行为识别,这些识别大都是穿越区域,覆盖区域,并不是动作行为化序列的技术路线,穿越区域属于控制检测器的后流程方向,一般不会涉及到目标分类器.

在某些数据科学工作中,例如从数据海寻找一个坐在凳子的人,会用穷举方式寻找坐凳子,实现这种挖掘的工作也会使用目标分类器.具体做法为:在每张图生成随机 100 万个框框,这些框框的尺度和倾斜角度都会随机,然后再把图抠出来与目标分类器进行匹配.通常来说这都是比较专业的 CV 工作,其目的是绕开检测器,或则是借助目标分类器来快速生成相似性标注,亦或是寻找数据,在这类流程运行中,往往会调度大量 gpu 设备和数据,同时也会消耗大量时间来计算.数据科学总是在有限条件下不断的做数据挖掘,然后分类,标注,如此反复,今天人类的大模型成果就是在长期的数据科学工作中慢慢积累而出的结晶.

目标分类器的推理

Z-AI 体系的目标分类器推理深度都根据不同模型而定,总结下来,分了 3 个层次

弱推理

弱推理走动态输入尺度,例如汉字(40*40),手写体(60*60),都可以使用 GNIC+GDCNIC 模型,这类模型是整个 Z-AI 体系中训练最快的模型,一般几分钟就能完成训练.弱推理的额定最大分类 10000 个.

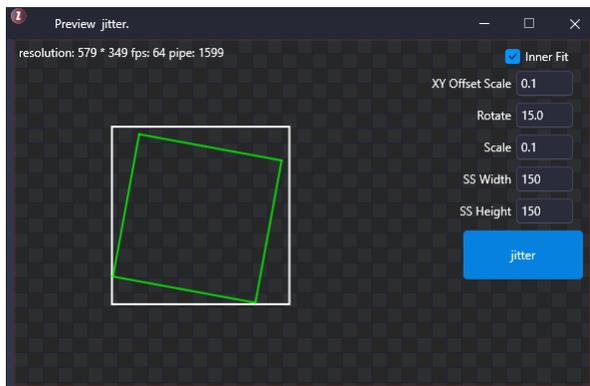
弱推理是直接把框框抠图,然后输入训练,会在抠图中做额外的色彩变化处理,不会做 jitter 这类采样处理.

中推理

中推理多用于度量化模型,同时 Metric 度量化模型没有最大额定分类的限制,另一方面度量化模型永远返回度量化向量,这时候,必须从线性方向额外计算向量相似度才能得到分类结果,而 GNIC+GDCNIC 则是直接返回分类结果.

Metric/LMetric:通常会作为人脸目标分类器使用,主要工作是识别目标身份,这类模型的输入尺度是固定的 RGB 彩色数据,例如 Metric 要求输入必须是 150*150,LMetric 则要求 200*200,对人脸抠图来说,150*150 的方框已经足够含有面部特征. **Metric/LMetric 分类器在推理时会向内做 jitter 采样**,如下图:在每个训练步数,会随机抖动从源样本采样去做 loss 拟合计算.

(Metric/LMetric 不光是输入尺度,返回向量长度也不同,Metric 返回 32 个双浮点,LMetric 是 24 个双浮点,更长的双浮点意味着梯度差更小,并消耗更多内存,反之,更短的双浮点梯度差会更大,消耗内存更少,更适合记忆 1000 万分类这类大数据流程)



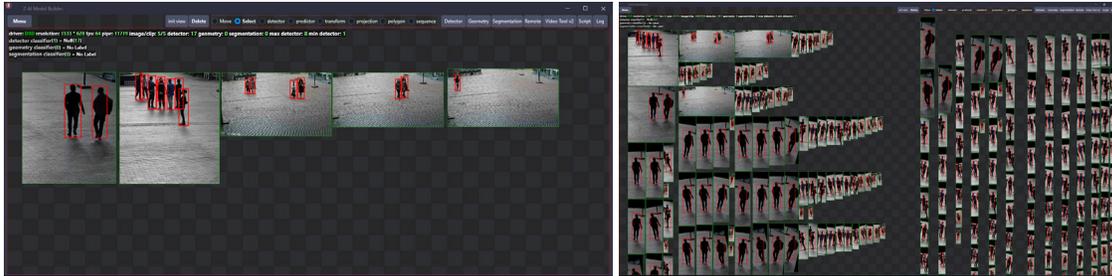
Z-Metric V1:简称 ZM1 这是在 Metric/LMetric 基础上改进的动态尺度模型,它不再被 150*150 这种固定尺度所限制,另一方面,ZM1 的输入条件也从 RGB 彩色变成了灰度,同时,ZM1 必须在等边条件下才会启动 Jitter 推理,如果非等边,例如 75*150 这类竖框会是弱推理.ZM1 之所以存在和被应用是因为它可以训练动态尺度,这意味着常用的人类,机动车,这类目标可以直接入手目标分类而不需要再跑去修改 DNN 的卷积网络.

强推理

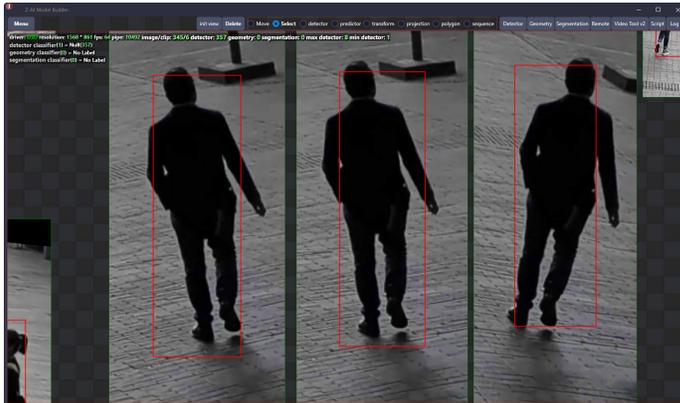
Z-Metric V2:简称 ZM2,这是在 ZM1 基础升级而来的强推理,也是目前版本唯一支持强推理的目标分类器.

ZM2 强推理器的由来是处于解决人类肢体动作识别时咬牙做的,第一代肢体行为识别使用 ZM1 的体系,当初针对 ZM1 体系给了一种铺数据的规则:就是针对目标框框,以几乎无穷尽的方式增加样本规模,这是一种程序化增容的做法,例如有 3 个样本,每个样本各有 2 个标注框框,这时候,程序化可以把 3 个样本增容成 3000 个样本,6000 个框框,依次类推,然后建模 ZM1,当时测试下来的准确度结果差不多为 97%.但这样干,对于运行建模以及使用程序化生成的工作站/服务器要求很高,普通的 normal 训练根本不可用,因为样本规模太大,框框太多,必须使用 Large-Scale 方式来训练(使用硬盘来暂存大数据,给内容腾挪空间),另一方面,因为 ZM1 建模中会有一个反推环节.当时,总结下来,大数据+超级耗时的反推,这让 ZM1 的训练效率不太理想,最后一咬牙就直接设计了 ZM2.

左图原始样本,右图为 ZM1 的增容样本



把样本放大一些会看到程序化增容会给框框随机旋转,缩放,镜像等等方式采样,然后重新生成样本,专用于目标分类器训练,等同于在建模标注工具层面直接解决将弱推理和中推理升级成强推理.



同理,所有的弱推理和中推理模型,在经过建模工具的样本增容以后,都可以是强推理计算出的模型,这时候,对训练服务器的配置就会变高,样本规模会往无穷大方向走.

ZM2 直接从底层算法角度,独立设计了一种模型,用于替代样本增容的做法,如果说 ZM1 的极限可以生成百万标注的样本,那么 ZM2 则是真正的无穷尽,ZM2 是在训练中引用增容的生成算法输入来训练,这样的目标分类器推理做法就是强推理.并且对硬件配置要求不高.

目标分类器的数据规格

首先理解分类器作用是给目标分类,所以分类器的数据就是带有分类的框框或小图.分类器并不像检测器那样对场景有所要求,分类器只需要确定框框和小图是正确的,接下来就可以训练分类器了.整个 github 的目标分类器几乎 99%的数据规格都是小图.

Z-AI 体系中除 ZM2 以外的所有目标分类器的数据规格都是小图.

训练分类器的数据前置处理

Z-AI 体系的所有训练数据就是样本库,并不会针对模型跑去生成特定的数据规格.

Z-AI 体系中的不同模型在训练时会对样本库做数据前置处理,这类前置处理会把样本库转换成小图(所有的分类器都是小图规格),只有 ZM2 才会使用带标注框的图,其余模型会把带标注框的图转换成小图.小图的转换规则为[拟合框抠图](#).

ResNet-Metric

训练参数中的 snapshot 为 False 时样本库的所有标注框会被转换成 150*150,当训练参数中的 snapshot 为 True 时直接把整张图片作为小图输入,如果图片尺寸不是 150*150 将执行强制 Zoom.

Large-ResNet- Metric

训练参数中的 snapshot 为 False 时样本库的所有标注框会被转换成 200*200,当训练参数中的 snapshot 为 True 时直接把整张图片作为小图输入,如果图片尺寸不是 200*200 将执行强制 Zoom.

GDCNIC

训练参数中的 snapshot 为 False 时样本库的所有标注框会被转换成 SS_Width*SS_Height(训练参数),当训练参数中的 snapshot 为 True 时直接把整张图片作为小图输入,如果图片尺寸不是 SS_Width*SS_Height 将执行强制 Zoom.

GNIC

训练参数中的 snapshot 为 False 时样本库的所有标注框会被转换成 SS_Width*SS_Height(训练参数),当训练参数中的 snapshot 为 True 时直接把整张图片作为小图输入,如果图片尺寸不是 SS_Width*SS_Height 将执行强制 Zoom.

Z-Metric-V1

训练参数中的 snapshot 为 False 时样本库的所有标注框会被转换成 SS_Width*SS_Height(训练参数),当训练参数中的 snapshot 为 True 时直接把整张图片作为小图输入,如果图片尺寸不是 SS_Width*SS_Height 将执行强制 Zoom.

Z-Metric-V2

ZM2 不支持小图,样本中必须有图片+标注框,ZM2 是在训练过程中动态生成小图,小图的数据尺度由 jitter_ss_width*jitter_ss_height(训练参数)决定.

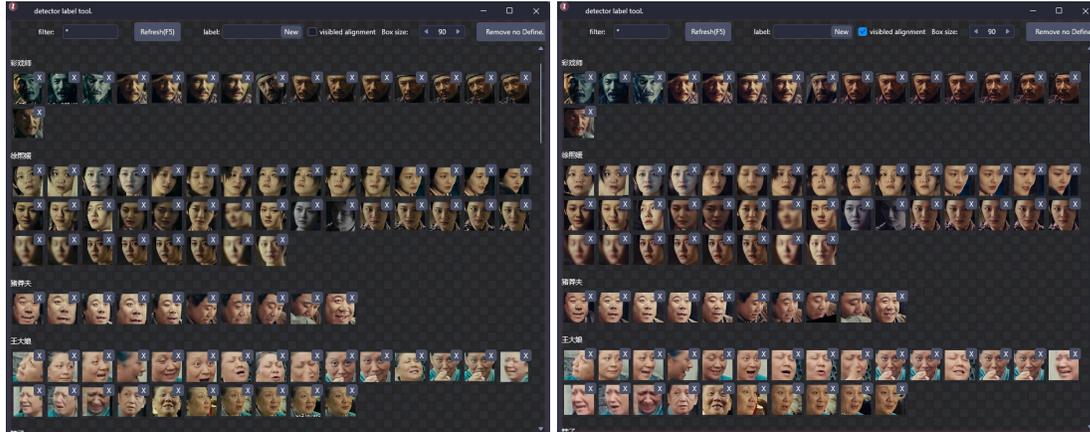
注意:ZM2 的前置处理不是生成小图,而是直接把整张图+标注框作为训练数据.

Prepare Rasterization 小图数据

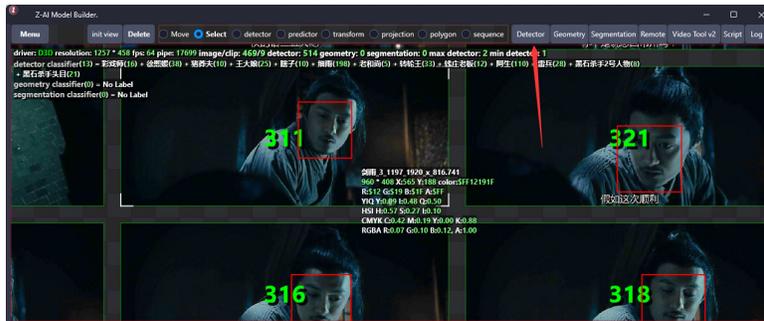
PR 是一种内置的小图规格,每个标注框都会带有 PR 数据,它的存在作用就是给训练器提供专用小图,从而替代框框抠图,尤其在需要走 Shape Predictor 模型的方案,例如人脸模型。

小图在标注工具也是以 PR 定义,例如人脸数据中的小图是指对齐后的人脸,对齐的意思是倾斜人脸或则侧脸将会做正脸化处理,然后生成 PR,这时候在训练时会优先以 PR 小图为数据规格.在训练的前置流程中,如果没有 PR 数据,那么将会以框框图作为小图。

下列左图为框图,右图为 PR 小图

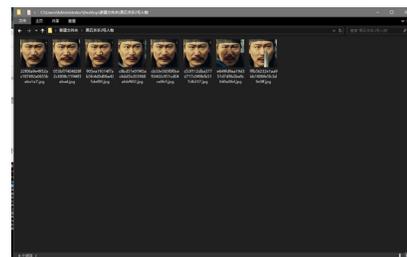
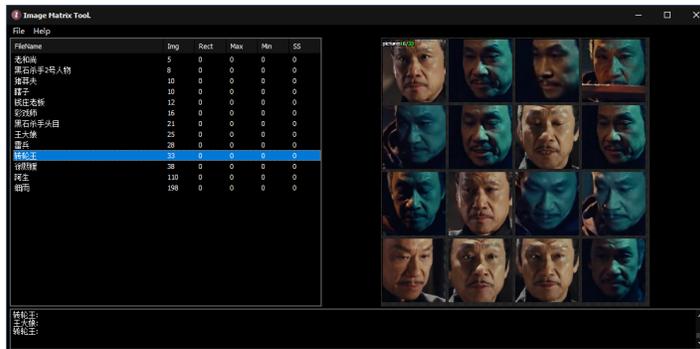


在 Model Builder 中必须使用检测器工具才能看到 PR 小图



在 Model Builder 也可以通过 Export 工具直接把 PR 小图按标签导出到某个目录中.同样的,也可以使用 Prepare Rasterization Tool 移除小图数据,当移除小图以后,将会以框框图作为小图。

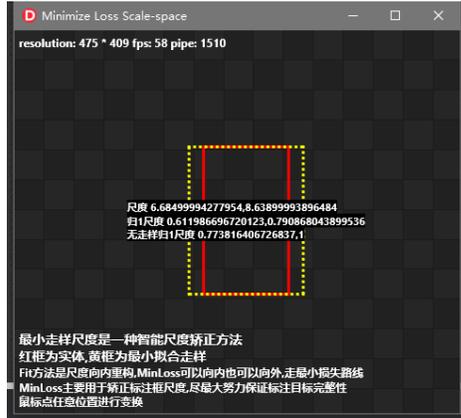
小图在图矩工具会是个体的小图片,小图在系统是规格的图片文件



拟合框抠图机制

ZM1/ZM2 系列模型训练时需要指定尺度,例如 100*300,那么尺度就是 1:3,如果框框尺度不是 1:3 数据就会做框框拟合处理,会从框框周围抠图做到 1:3

下面是 Z-AI 体系中的一个标准 Demo,演示了拟合框的尺度变化,整个 Z-AI 体系对拟合框的计算都不再是对内拟合,而是内外相结合,取最小走样尺度做拟合.当框住的目标不符合框尺度,无论内部转换,导出小图,传递,都会以最小走样取拟合框.这必须在 Z-AI 1.4 或则更高的版本才会这样干,在早期版本拟合框都是对内拟合,例如长框向方框拟合会导致框内缩.



生成 Prepare Rasterization 小图

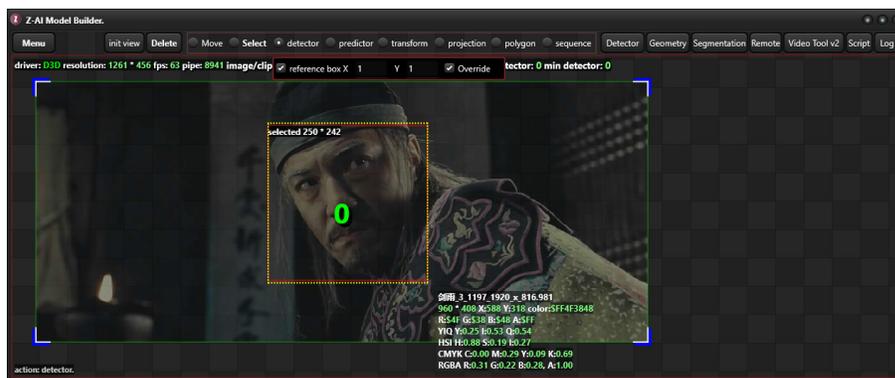
PR 小图本质上是一种数据,用于表达标注框的数据:标注会是一个框,PR 则是从框里面生成的数据,而最终,PR 小图会被目标分类器使用.简单来说 PR 是一种数据生产上的技术.

PR 小图的生成依赖于 Shape Predictor(SP)模型,SP 模型用于描述对象形态,输出 Shape 几何数据,当有了 Shape 数据以后,就可以做计算处理了,例如用 68 这类顶点表示人脸模型是根据鼻梁和下巴以及双目横距来决定生成正脸的 PR,而 4 顶点则用于矫正歪斜目标提供干净 PR.

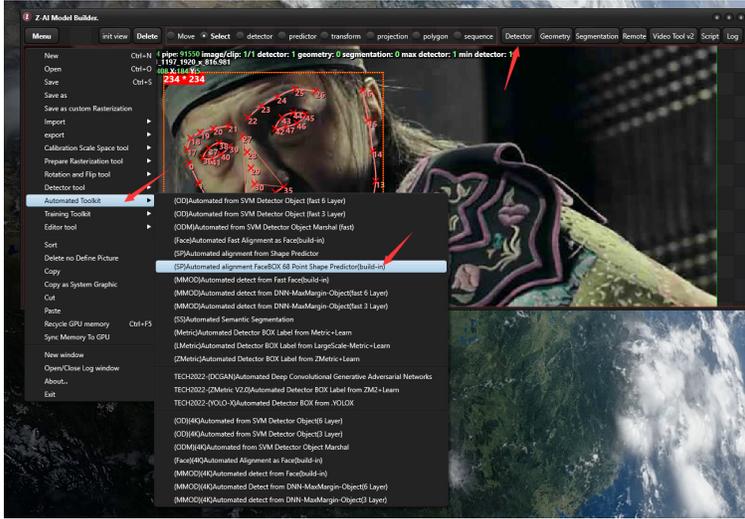
196 人脸面部顶点通常用于 3d 面部重建,2d 转 3d 需要必须具有描述面部菱角信息的数据,只有这样程序才可以线性化重建 3d 面部.本文以目标分类器为主,对 SP 不做过多讲解.

通过手动画框来成 PR 人脸小图

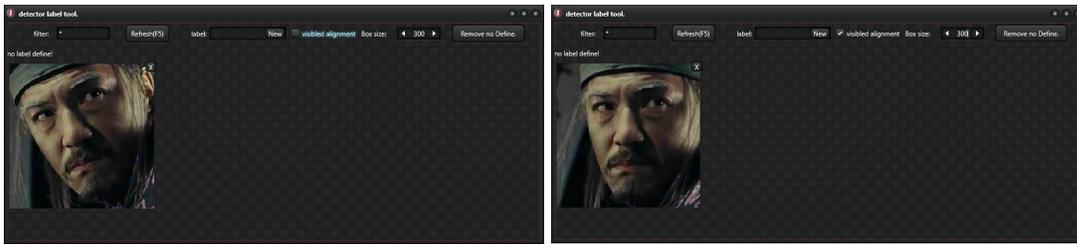
首先切换到检测器标注,打开参照尺度,比例给 1:1(这是人脸模型尺度),然后,框住一个人脸



然后使用内置人脸 SP 模型自动化标注,这时会生成 68 个面部五官几何坐标.当看到人脸坐标,PR 就已经生成完成完毕了.这时候,再通过 Detector 工具可观看 PR 结果.

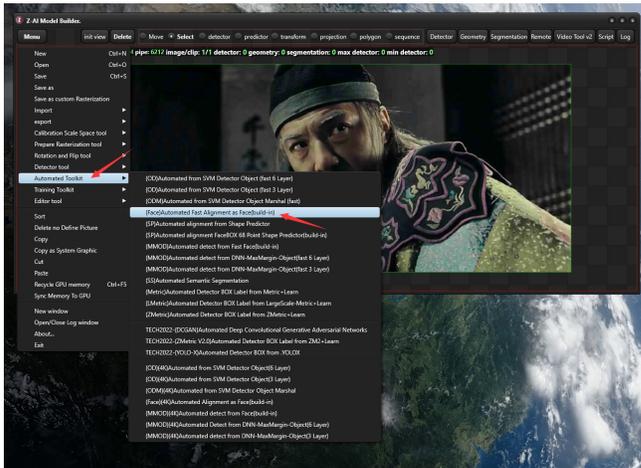


左图为原始标注,右图为正脸,而这张正脸就是 PR 数据.



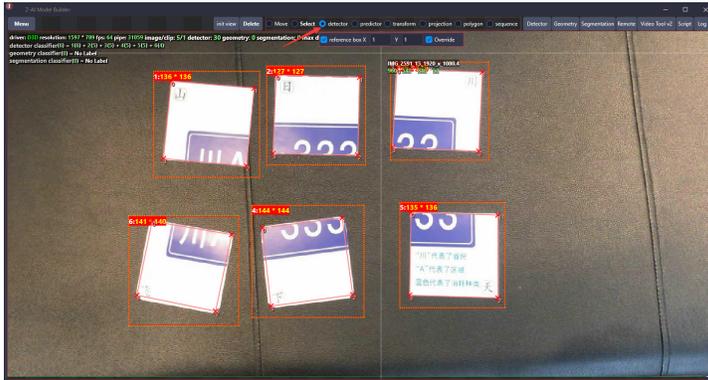
通过人脸工具自动生成 PR

不需要自己拉框,它会自动检测人脸,然后再运行 SP 模型,最后直接生成 PR,然后,直接给人脸打标签训练就能得到人脸的目标分类器,也就是,人脸身份识别的模型.



通过手画 4 顶点生成 PR

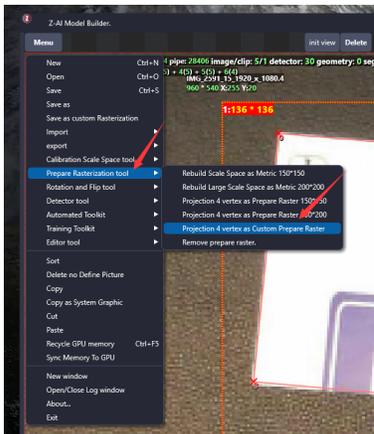
首先,切换至检测器框框标注,给目标拉框



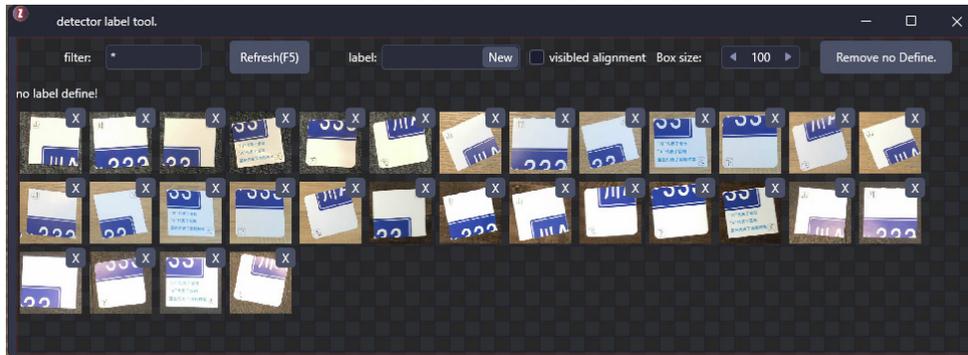
然后,再切换至 SP 标注,依次从左上,右上,左下,游下,点 4 个坐标,表示目标形态



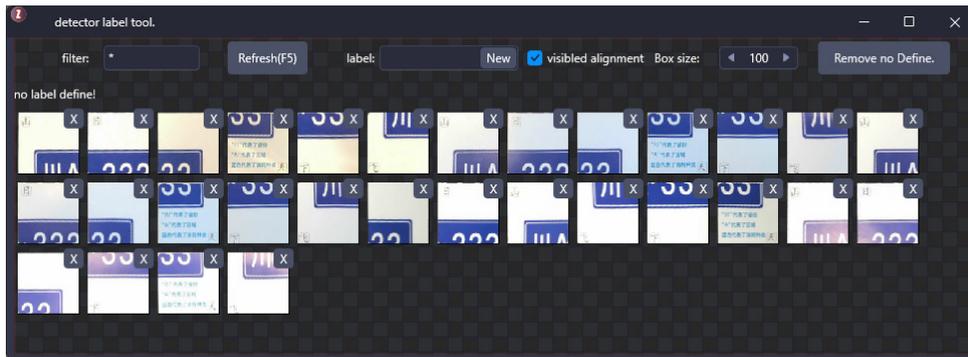
然后切换至 PR 工具,使用 4 顶点投影新 PR,在弹出参数中给 PR 的尺寸,接下来生成 PR



当 PR 生成完毕,通过 Detector 可看结果,这是标注框的形态



使用 4 顶点生成后的数据



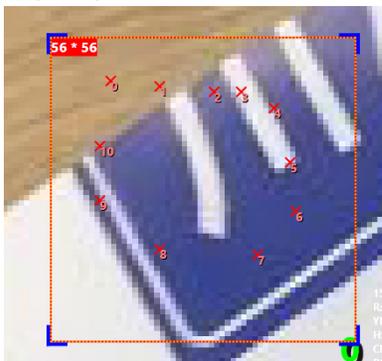
4 顶点的 SP 模型

应用 4 顶点 SP 通常会针对 4 顶点建模一个专用 SP 模型,然后再用 SP 模型针对框框大批量来生成 4 顶点,最后,是修正错位的 4 顶点,接下来,就可以直接生成 4 顶点的 PR 了。

通常来说,在固定光照和环境下,SP 模型在工业视觉领域表现会比较突出,例如抓取定位,但工业并非简单建模 SP 这种事情,SP 模型同样也需要 Runtime 系统来驱动业务运行。

SP 建模以及 SP 的训练参数,会有专用文档。目前标注体系目前只能支持 68 顶点人脸与 4 顶点,如果自定义例如 10 顶点,会没有轮廓线,这将让标注难以排错,因为 10 顶点的轮廓图形描述是少有的情况,通常来说,只要是常用的 SP 标注,就会给出图形描述。

下图是个 10 顶点的 sp 坐标,因为不知道 10 顶点描述的目标形状,因此标注体系无法给出图形参照信息。



目标分类器的反推机制

反推是把识别的结果反推到样本位置,反推可以精确具体的标注框,或则小图.因为当目标分类器堆大以后,样本数据修正将会是必须的工作,重建一套目标分类器数据库会等同于重建大数据结构,目标分类器的大数据在应用后期只能走修正路线,而修正的方法就是用反推.

反推是一种机制,目前反推支持的目标分类器模型:Metric,LMetric,ZM1,ZM2,除此之外的模型像 GDCNIC,GNIC,ResNet-Scene(RNIC),Large-ResNet-Scene(LRNIC)这类模型都是不支持反推的.

反推在训练流程中会先走正推,以 Metric 模型为例,当 Metric 训练后,流程会输入当前所使用的训练样本,从而得到 Metric 模型所输出的向量,例如张三,会有一堆向量数字,这些向量由 Metric 模型提供,这时候,对向量的处理工作会是存储,载入,搜索,在 Z-AI 体系中这是通过 Learn 引擎来干.当使用 Metric 模型对当前样本库生成向量时,会附加一条输入的样本信息,这条信息会包含样本反推信息,身份信息,分类信息,然后随着向量一并保存在 Learn 引擎,当正推流程全部完成后,训练流程会保存.Metric+.Learn 两个模型文件.例如在人脸识别时,.Metric 模型负责返回人脸的向量,.Learn 模型则是提供最接近该向量人的身份信息,同时.Learn 模型也会提供与目标身份所对应的样本库,标注信息,小图信息.

保存向量为什么使用 Learn 而不是 KDTree 这类二叉树结构

在早期项目中有一些人脸需求:解决高准确率的人脸识别问题.

这时候,Z-AI 给出了一种叫做候选机制的做法,候选机制就是,目标只要达到一定条件,那么目标就即是张三,也可以是李四,数据输入会来自一连串图像,这些图像大都来自实时视频,手机,IOT 终端这类设备,这时候会形成一条数据链,而数据链会包含张三,李四,王五,最后,会通过统计来得到最终的结果.简单来说,就是每张图都会有一个人,这个人的识别结果是候选机制的,会有许多符合条件的匹配结果.

回到小节,KD-Tree 这种东西是快速搜向量,它的机制是返回单一化向量搜索结果,并且这还不一定会是最接近向量,因为 KD-Tree 是从中间切.Learn 对向量是最优搜索排序化结果.这不是单纯而暴力的循环,而是使用了大量结构体的优化搜索.

另外一点,只有 Learn 可以存标签和反推信息,KD-Tree 只能存一个标签.

在良好的 Metric 模型基础上,Learn 的第二次建模

对人脸来说,Metric 是面部数字化模型,Learn 是数字化标签处理模型,Learn 的第二次建模是直接再次输入新的人脸+身份来建模.

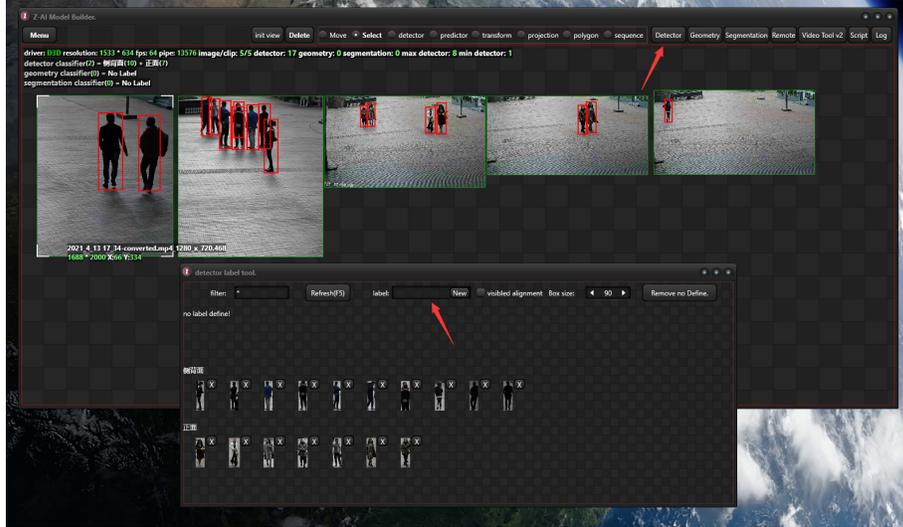
Learn 的第二次建模是一种机制,只需要训练一个规模稍大的 Metric 模型,然后围绕该模型不断的给新的人脸+身份,存库后就能做查询了,Learn 计算 1 万人大概只用 5 秒,而 Metric 计算 1 万人,至少需要半小时.实时的机制对应用层来说还是比较重要的.

Learn 二次建模通常会做实时更新模型,例如当天记录面部状态,然后立即用 Learn 二次建模,一直走迭代路线.

在 Model Builder 标注目标分类器

通过 Detector 打开标签工具,使用拖动操作即可完成标注

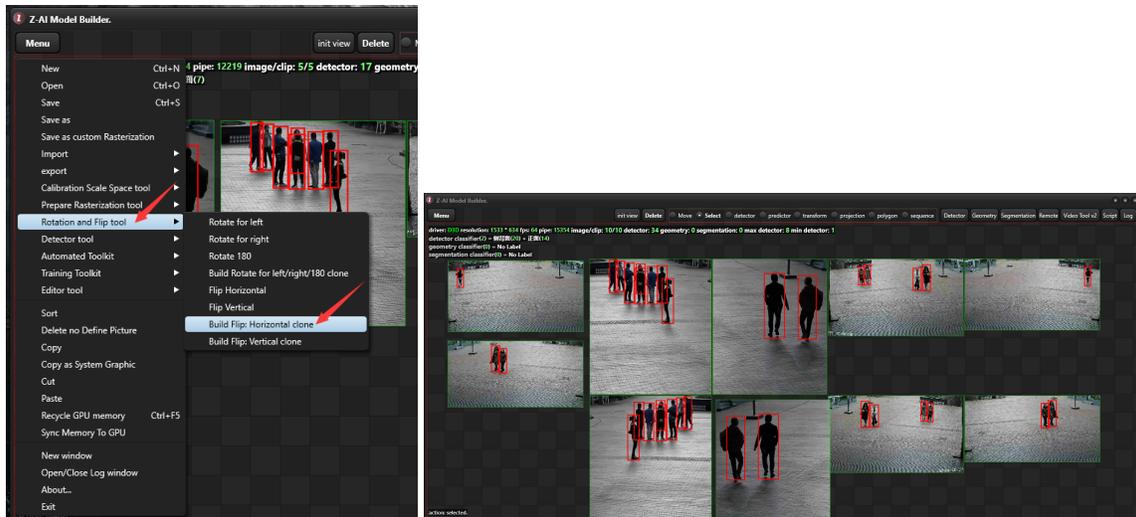
目标分类器与检测器不同,目标分类器一次只能一种尺度,例如全都是行人,这时尺度是 1:3,不可以再出现 1:1 这种方形标注。



当标注完成,亦或 PR 小图是正确的,这时候,就可以开训练任务了,目标分类器与检测器不同,目标分类器的训练 loss 通常会非常的低,大部分模型会低于 $1E-5$,甚至 $1E-10$,并且训练会非常的快,训练上千个目标分类只需要 1-10 分钟,如果使用训练样本测试往往是 100% 的准确率.因此建议目标分类器在建模和测试时,训练用一个样本库,测试用另一个样本库。

目标标注镜像

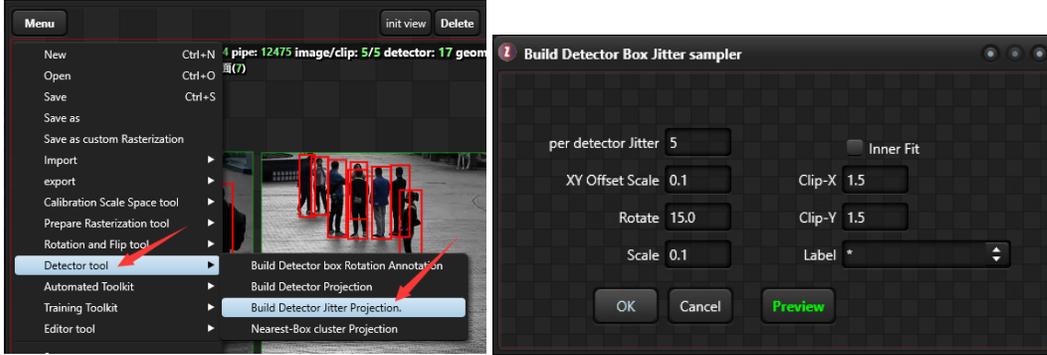
当使用弱推理的目标分类器模型时,可以做一次横向的镜像,这样可以加大样本的多样性.镜像并不会破坏标注尺度,以及标注中的 PR 小图,同时也不会破坏 Shape Predictor 标注数据



目标标注抖动

抖动标注是使用弱推理模型亦或中推理模型,需要更多样性的采样

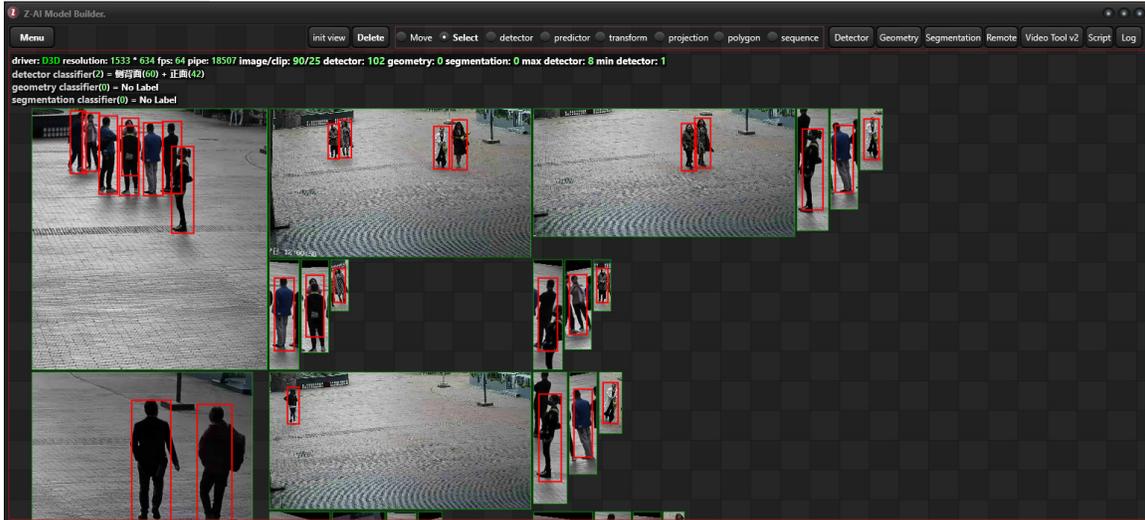
下列参数会对每个标注框随机抖动 5 个新样本



这时候可以留意下目标分类器的计数器,被抖动以后,这些计数器会增加数倍数量.



抖动以后的标注框采集角度会更加的丰富.抖动并不会破坏标注框尺度,PR 小图,SP 标注信息.当使用 Metric/LMetric/ZM1/ZM2 这类模型时抖动数量不宜过多,度量模型有反推,正推,候选化这类机制,如果抖动数量过多,例如超过 10 万,这时候模型运行中的正推会非常耗时,以及导致.Learn 异常庞大,而当目标分类器运行于业务系统时,超过 10 万数级以后,候选化机制将会发生超延迟,计算一个目标的候选分类可能会多达数秒,而目标分类器正常工作延迟应该达到每秒数千次计算.使用抖动功能一般建议保持在 5 万目标以下.更加丰富的样本来源会真正让目标分类器具有多样化识别能力,算法角度可以锦上添花,但多样性的样本源会是一种质变提升.



ZM2 的目标分类器

ZM2 具有强推理能力,对 ZM2 来说,PR 小图,人脸对齐,抖动,镜像,全都可以省略掉.在本文的数据规格环节,对目标分类做了大量机理和技术层面的描述,这些技术,全都是建立在弱推理和中推理的条件下,各种操作错综复杂.而一旦选择使用 ZM2 作为目标分类器以后,就不需要再纠结数据规格问题了.**ZM2 的训练流程是目前整个 Z-AI 体系最先进的,就目标分类器的识别结果而言,同样的样本和标注,使用 ZM2 无论从建模,训练,识别结果,综合起来是最好的.**技术更新迭代的结果,这是从另外一个维度出发重新设计的目标分类器,实用性更高.

ZM1/ZM2/GNIC/GDCNIC 这 4 个模型都有个共同缺陷:因为需要动态尺度计算流程,导致的目标分类器在 DNN 数据传递上走的灰度矩阵数据,简单来说 ZM2 模型可以看见的目标是灰色,不是彩色.支持彩色的模型目前只有 Metric+LMetric.

注意:**ZM2 只能训练带有标注框的图,不可以是整张小图.**

检测器使用 DNN-OD/YOLO-X,目标分类器使用 ZM2 是目标比较典型的搭配组合.

在图矩工具用大数据标注目标分类器

图矩是数据集的集合,一个

在后续文档关于肢体行为模型,场景分类模型,也会使用到图矩工具,用稍微概括一点的话来说,使用图矩等同于在鼓捣大数据领域.

在 AI 的发展历程中,算力越来越高,数据就会越来越大,在这时候,有一些模型不支持大数据,例如检测器使用大数据训练,loss 往往难以走出让人满意的结果,而**目标分类器面对大数据会表现出非常好,目标分类器+大数据具有发酵能力,当数据多了,目标分类器会变的非常聪明**,这时候,会有许多可以挖掘的东西,例如,把目标框翻译成文字信息,当图片以监控系统或则以手机视频的方式工作,那么就可以是信息海,这些信息可以被人为的利用和挖掘.做出这类系统的前提是把目标分类器大数据化.

在图矩中堆砌目标分类器样本

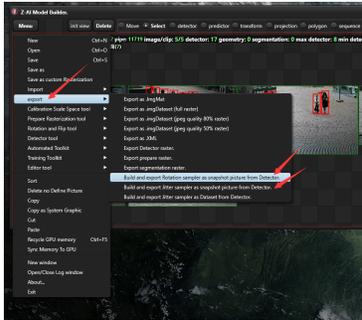
在图矩堆砌样本需要走规格路线,目标分类器的规格会错综复杂,可以小图也可以标注框,并且这些小图与标注需要与目标模型的算法相匹配,例如 Metric 模型会要求等边,ZM2 模型会要求训练参数对应标注尺度.只要坚持规格路线,就可以机械式的通过标注工具来生产数据,然后,在图矩中集成:直接导入.AI_Set 文件.

图矩工具中的尺度系统

在图矩工具中并没有 Model Builder 里面的尺度矫正功能,因为尺度矫正是一种对数据的修正.图矩的数据思路是堆砌,图矩面对尺度差异问题,建议直接走移除路线.图矩工具没有统一化的尺度处理,最好办法是人工分批把数据处理好,然后再集成进来:直接导入.AI_Set 文件.

在 Model Builder 生成小图并导入图矩

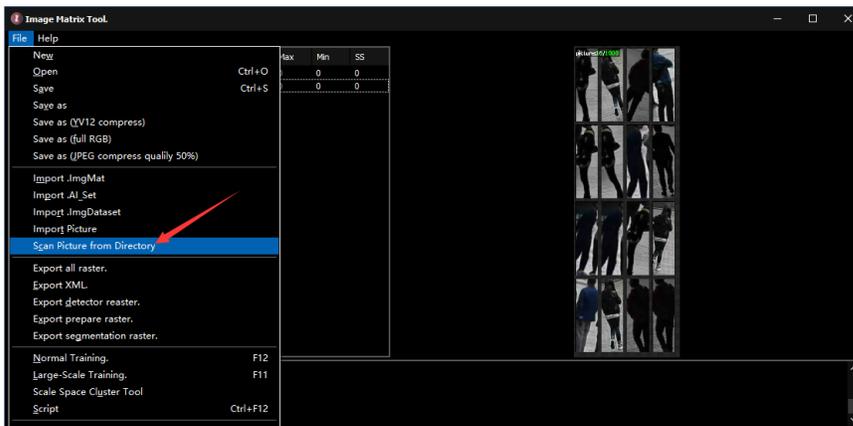
生成小图前需要先做标签工作,确保所有的框框都加了标签,然后使用导出工具来干,下图菜单分别是导出旋转小图和抖动小图.小图可以支持除 ZM2 以外的所有目标分类器模型.



小图导出会弹个抖动参数,小图尺度,当导出完成后,会在目标目录形成带标签的子目录.

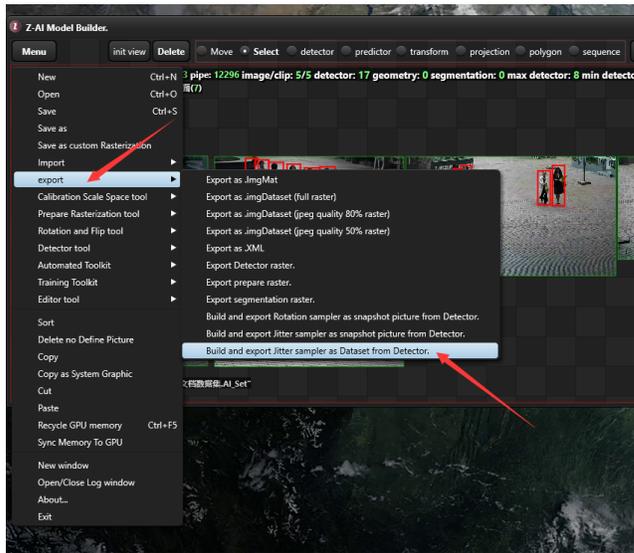


这时候在图矩直接再导入整个目录,这时候小图数据也就出来了.

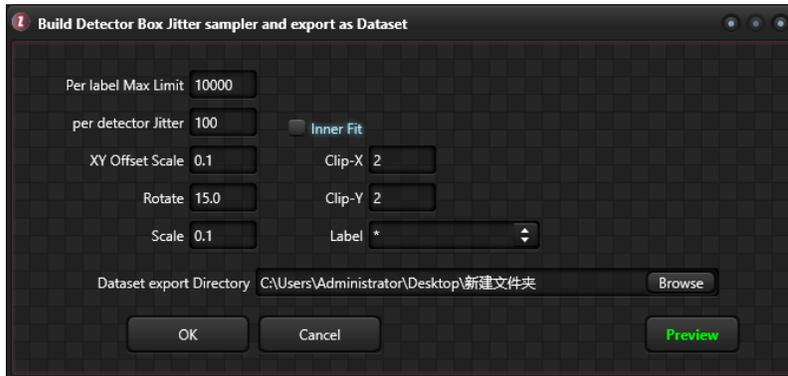


在 Model Builder 生成带有抖动和标签分类样本库并导入图矩

进行该操作前先用标签工具做好标签分类,以及尺度统一化,然后按下图菜单启动导出工具



在参数窗口中,10000 表示每个标签分类的最大数量限制,如果超过 10000,会做随机剔除处理,100 表示每个标注框生成 100 张新图并且抖动.

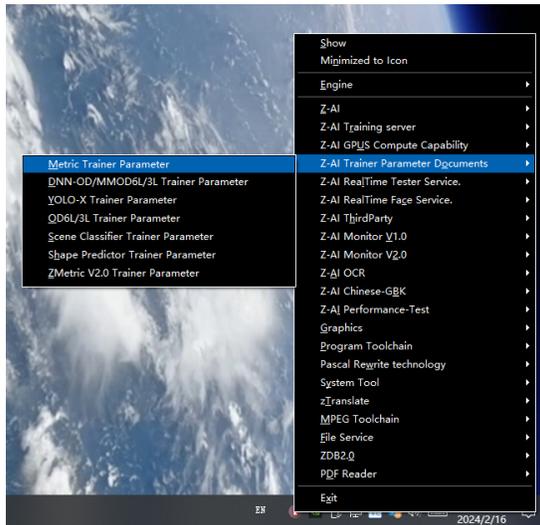


待导出完成后,会根据生成目标样本库,可以直接导入导图矩,带有标注框可以支持 **ZM2 模型**

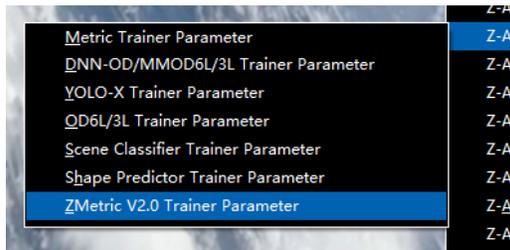


训练目标分类器

Metric/LMetric/ZM1,这三个模型的训练参数参看 Metric Trainer Parameter 文档即可



ZM2 模型的参数会复杂许多,有 ZM2 的专用文档,使用 ZM2 务必要明确一点,ZM2 不支持小图训练,必须是带有标注框的图.



最后是 GDCNIC/GNIC 这两个模型,这两模型目前没有参数文档,只能从标准 Demo 获取那些参数的信息,一般来说,参考 Metric 模型的文档也是可以的,GDCNIC/GNIC 的训练参数很少.

在所有目标分类器的训练流程中,只有 ZM2 可以使用 2 个 GPU 高效率训练,其余模型单 GPU 会高于多 GPU.见来说,只有 ZM2 可以开 2 个 gpu 得到高于单 GPU 的训练效率.根据不同的 cuda 版本和准系统而定,某些准系统训练 ZM2 可以使用 2 个以上 GPU,某些最多只能开 2 个.通常来说大部分目标分类器在 1 小时内都可以完成训练.

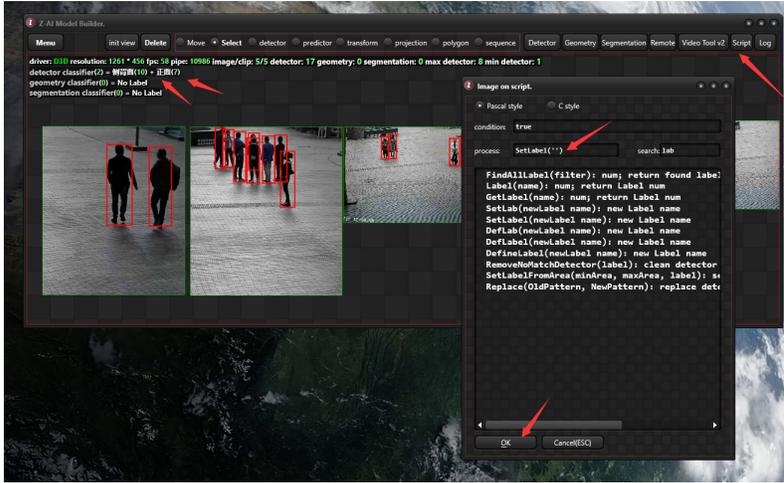
Loss 环节:目标分类器与检测器不同,当标注的目标有许多相似点,只有一些微小的地方存在差异时,这时候,Loss 必须非常小,Loss 精度至少应该走到 1E-3,既 99.9%以上,只有这样的精度才可以让目标分类器准确计算结果,如果训练完成时 Loss 走出 90%这种精度,这会是数据不可区分,这时候应该是样本中存在相似性的数据,既不同分类间存在高度相似性的数据.如果数据繁多,导致修正反人类,第一条路是直接走测试流程,第二条路是重新做一下样本.

测试目标分类器

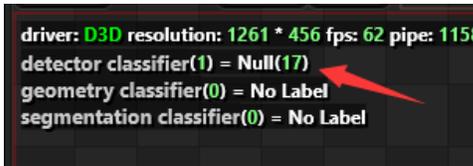
简单测试

如果目标分类器建模走的是小数据流程,10000 个样本,直接使用 Model Builder 来测试,进行测试的样本可以非训练样本可以是训练用的样本.最简单直接的测试直接用 Model Builder.

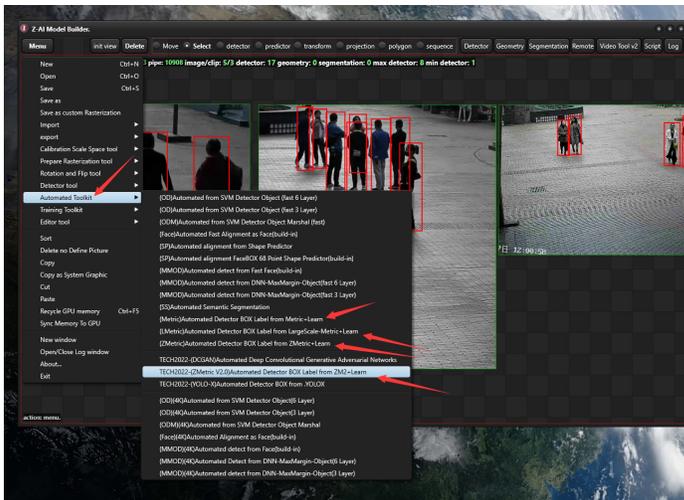
测试方法:先检查标注框的样本是否标签,如果有标签,使用脚本 SetLab(""),把标签全部清理掉



完成清理后,这里会看到空标签信息

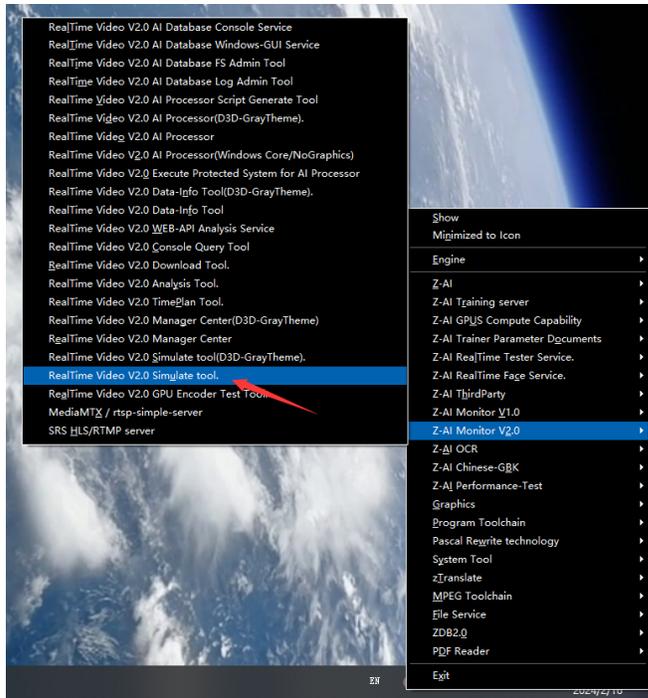


然后使用自动化标注工具:箭头所指含有 Metric/LMetric/ZM1/ZM2 一共 4 种模型,直接肉眼对比识别结果即可,而对于 GDCNIC/GNIC 这两个模型,无法直接测试,需要自写测试程序.

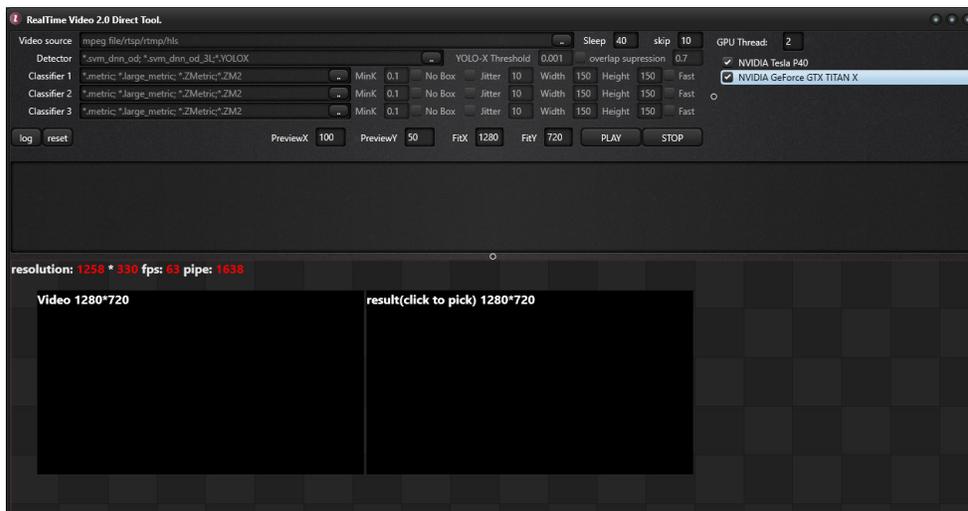


大数据模型测试

如果目标分类器的规模很大,这时测试应该走业务仿真路线,通过工具链菜单打开 6 代监控的仿真工具

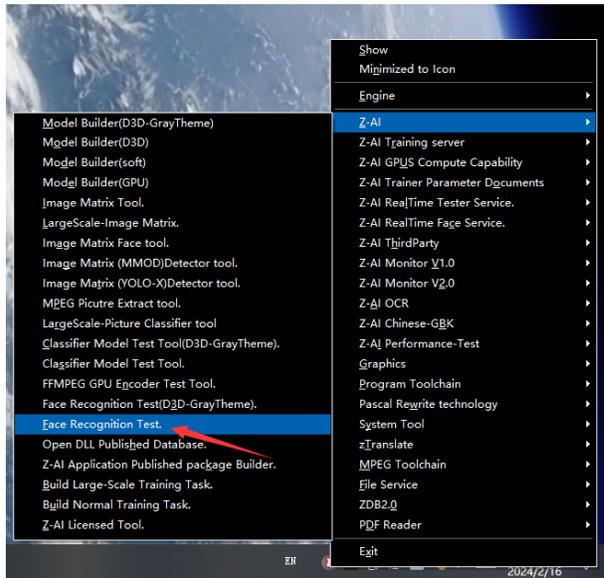


使用仿真工具测试需要使用视频数据源+检测器+目标分类器,只有让这 3 个数据到位才能做目标分类测试.最后的测试结果用肉眼观看就行.支持候选识别机制.

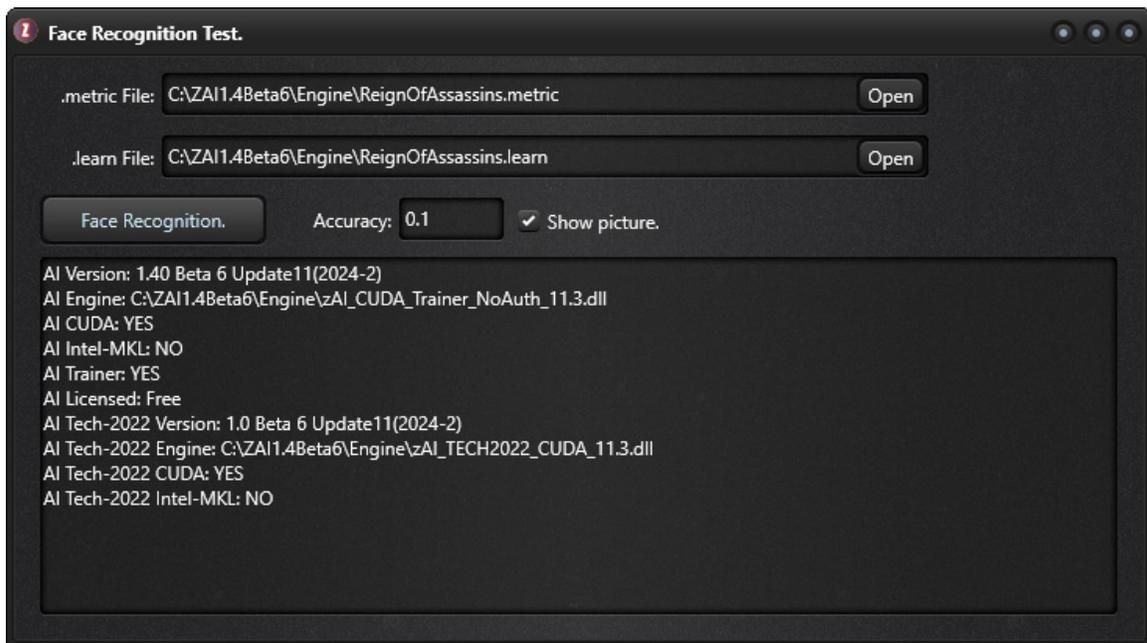


测试人脸模型

如果目标是人脸模型,使用人脸测试工具,只能支持 Metric 模型



该工具只能以图片作为测试数据源,返回结果是唯一化模式,没有候选支持



业务系统中的目标分类器

目标分类器是一种技术产物,技术没有产品属性,目标分类器只能作为产品中某个流程的框框图识别环节.

把目标分类器应用于业务系统通常会放在检测器和坐标锚的后面,这要从整体来考虑,例如检测器找不到目标,那么目标分类器就会无法工作.例如目标分类器的输出信息非常多,随便一个小框也可以输出几百条候选信息,如果把这些信息通过有效统计出来.这一切都是一种趟坑,需要从整体来考虑.

如果从整体来考虑,那么目标分类器的局部任务就会很明确:无论挂接到哪种型号的 GPU 平台,目标分类器必须可以有效发挥硬件潜力.其次就是编程基本功,很多 lead 和客户,他们有许多美好的想法,而就落实 AI 到应用这一层来说,其实,这是编程基本功,一切都需要围绕计算机机理来进行,美好的事物,以及由巨大利益诱惑的事物,切勿被它们冲昏头脑,美好的结果只有在实现以后才能达成,并且不会是一次性的实现,凡是产品均会迭代升级,这需要掌握一些技术点.对此,目标分类器在业务环节提供了各种技术面的方案,例如 6 代监控框架,RealTime Tester 框架,Rasterization Queue 框架,Face 框架,这些框架是面向技术的,如果需要业务应用,躺坑会是命中注定的.

Z-AI 的整个技术栈体系比较纵深,正常情况下,负责业务系统这个环节,应该委托作者帮忙,或则寻找一位可以胜任底层机理和算法的技术员来做业务系统实现,这可以对已有框架自己修改,弄成业务用的目标分类器.

全文完

By.qq600585