Z-AI1.4 版场景模型建模

文档版本:1.1

最后修改时间:2024-3-3

修改日志

2024-3-3 第一版本撰写完成

2024-3-4 添加场景模型训练章节

目录

什么是场景模型

场景模型就是人们日常使用的照片分类器,因为照片分类器是个充满不确定的技术方向,当照片分类器使用大模型遇到私有化照片,结果会非常不好.而照片分类器的形式,技术层面的做法却是让人们非常认可,这非常矛盾.一方面识别结果不尽人意,另一方面又被广泛使用.

图像世界内容无穷尽,人类的做法是建立已知图像的大数据,这种工作是从局部向无穷尽探索中和前进中的大数据工作,例如,搜索引擎词条会默认带有图片分类,这时候,即使把 google 的全部词条+数据都拿来训练,也只能做到识别常用和已知的图像,无法适应无穷尽图像,在科学层面,问题复杂性必须对等解决问题复杂性.人类使用有限数据企图解决无穷尽数据的方法并不对路.换句话说,通用照片分类器可以等同于通用人工智能,用有限的数据解决通用问题我认为最后会失败,但这是伟大的失败,这会推动大数据技术,算力,以及人们认知的进步.这会极大的带动局部的大数据应用,这种应用会是一场革命,而革命的路线,行为上总是先个体行为再到集体行为,也许遇到无法解决无穷尽问题,然后,全人类像蚂蚁一样齐心,最终用社会主义集体力量解决无穷尽.

深入讨论图片分类器前,需要回顾一下分类技术的前世今生,经典的图片分类器是随机抠图,也就是小图技术,这跟目标分类器非常类似,当一张照片被随机抠 10000 次以后,基本上这张图可能包含的信息都会变成机器学习的目标,换句话说,这张图被学会了,当机器学会机器会把照片以数学的形态记忆起来:就是 DNN.这是人们非常认可的一种做法,因为数学形态是记忆图片的特征.当图片规模达到 100 亿,这时,会需要设计一种非常庞大的网络以此来记忆整个互联网的图片.另一方面,这也会需要庞大的计算能力才能做到把图片转换成数学形态.

在 2020 年出现过一篇 paper,"Barlow Twins: Self-Supervised Learning via Redundancy Reduction",这里就简称 SSL,当时看作者署名,发现尽然有 Le-Cun(一个不玩虚的科学家).我不确定 SSL 与 GPT,OpenAI 是不是有所联系,我非常确定一点:这套思路几乎可以算完全重塑经典抠图方法,这是从数学层面无穷尽的推理思路.经典抠图是随机抠,走已知路线学习,而图片中是会有未知数据的,SSL 是面向未知的,这些未知数据,会被数学化,保存在 DNN 里面形成母体,这时候,通过第二次深度学习,也就是反推技术,对母体进行已知挖掘,最终会形成可以应用的模型.母体这套思路,简单来说,先记忆数据,当有输入数据出现在记忆之外,就开始增加记忆,这种记忆是无关数据标签的,可以是任何数据,不需要标注,不需要 google,只需要无穷尽的数据来源记忆就可以无限的增加.当数学层面的计算完成后形成母体,经典的图片分类器方法,会在母体基础上做第二次深度学习,走反推路线,经典方法将会完全建立在母体基础上.按目前发展趋势来看,拥有算力的机构和公司会作为母体提供方.另一方面,假定母体面向的数据是 1 万张图,它的计算规模可以约等同于上亿经典分类数据,这 1 万张图会有很多未知信息.这时候,有了方法,算力将会是一个大问题,假定要在 100 亿图片的规模上建立母体,也许要全球的 h100 都运行起来.总结一下:**母体路线会无限贴近无穷尽,算力大约需要提升 10000 倍**.

图片分类是大数据的前置工作,前面有了分类,后面才会高层建筑.否则面对茫茫数据海整个大模型趋势会陷入停顿.在另一方面,**图片分类就是场景分类,这在应用层面可以提供子程序运行条件,例如,街道走机动车识别,走廊走人脸识别,白天 AI 干活,晚上 AI 休息,这会让程序看起来挺有通用性**.

场景模型的应用

场景模型的应用都是框架化的,它需要分支流程互相依赖,例如识别到日间那么就启用人脸识别,这并不是传统流程.因为 GPU 的处理能力远超 CPU,传统单线程模型无法带动 gpu,会造成花大把钱购买 gpu 设备而使用率只有 10%.许多后台使用 python 的 AI 服务器会一次开辟多个进程来提升效率,这种方式后面会有统一性管理的问题,会有许多配置资源,管理计算,管理计算结果这事情,这是非常复杂的管理工作.

场景模型的框架技术都是建立在 DNN-Thread 技术体系上的事件回调,DNN-Thread 会用多线程调度 cpu->gpu 工作,通俗来说:cpu 需要先发数据给 gpu 然后才启动计算,待计算完成后,再取出数据,这就是一个 IO 过程,多线程流程是把原本会走单管道模型的排队计算进行批次化处理,例如传统流程会一个算完再算下一个,DNN-Thread 的多线程模型则是 10 条线程同时算 10 个,DNN-Thread 在内部是抢占式计算,假如队列有 1000 个计算任务,10 条多线程模型会抢先拾取首数据来计算,这时 GPU 几乎会 100%满载,而 CPU,内存,硬盘,网络,也都会处于高负载状态.当场景模型需要走多级分支,当业务流程需要从日间识别到人脸识别,内部会是一个事件桥的方式,场景识别一张图,触发识别结果,判断条件,如果满足,把图片发给人脸识别.这一整个流程需要框架才可以完成,不可以直接引用 demo 去干,因为 demo 的作用是演示功能,在真实的数据中心 hpc 服务器会有配置差异,人们总是喜欢用最少的钱购买算力并希望让算力可以彻底发挥潜力,因此,框架不光需要完成流程还需要适应不同算力配置的硬件环境.

真实的 AI 业务运行,会使用大量配置和脚本来描述流程和管理算力资源,如果把系统做成固定程序,那么实际运行中,也许会针对不同的服务器和环境修改代码,重新 build 运行,这太累了.目前最佳的 AI 框架,能运行用脚本描述的业务流程,也能用脚本来管理算力,这样可以做到在系统集中成现写现用.

一个题外话,当框架大量使用脚本和配置文件解决业务问题时,在未来,框架就会往自动化方向发展,会衍生出写一张 excel 表,或则通过 ui 填写一些数据,整个业务流程即可如期运作.这是未来的事情,当前还是注重要点:在业务系统运用场景需要使用框架,避免去走 phthon 多开进程的路线,避免针对每台服务器去 build 程序.

多级场景模型识别

有一种场景识别是分级方式,先识别到日间或夜间,然后进入下一级识别场景是街道,门禁,还是走廊,这时候,如果是走廊那么就启用斜 45 度的行人检测模型,如果是门禁则启用人脸识别.

当上诉场景在建模和部署时,会存在日间夜间模型,以及街道+门警+走廊的模型,行人检测器模型,人脸身份模型.

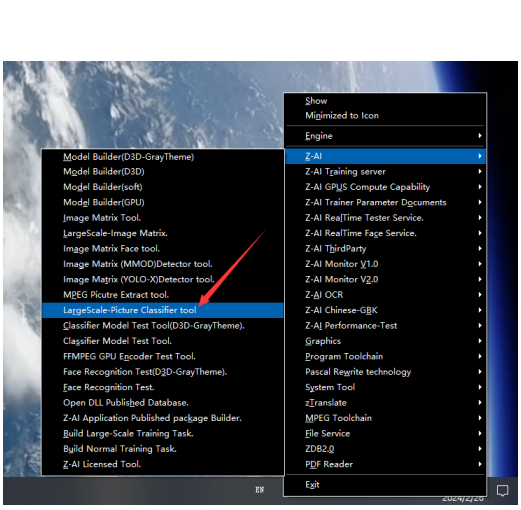
驱动这些模型的工作要依赖于框架.而这种框架会非常复杂,不再是普通的流程框架了,因为普通的流程框架只是跑通少量 AI 识别流程,当业务流程复杂化以后,框架就会需要适应多种不同的场景需求,而要让框架做到适应,做法就是往框架里面无限的堆 AI 程序,当堆到一定程度后,框架就会具备适应性,这种适应性会是一种被人们普遍抄袭学习的核心技术,或则说核心设计思路.目前我管这种模型叫做巨程序:当驱动 AI 的业务框架越堆越大,框架就能解决方方面面的需求,做到面对需求就是建模,写脚本,而不是写程序.无穷尽的巨程序是未来适应各行业 AI 识别需求的地基.因为未来任何的 AI 识别都不会 脱离场景识别的,而有场景识别就会有分支,有分支需求就会有巨程序的用武之地.

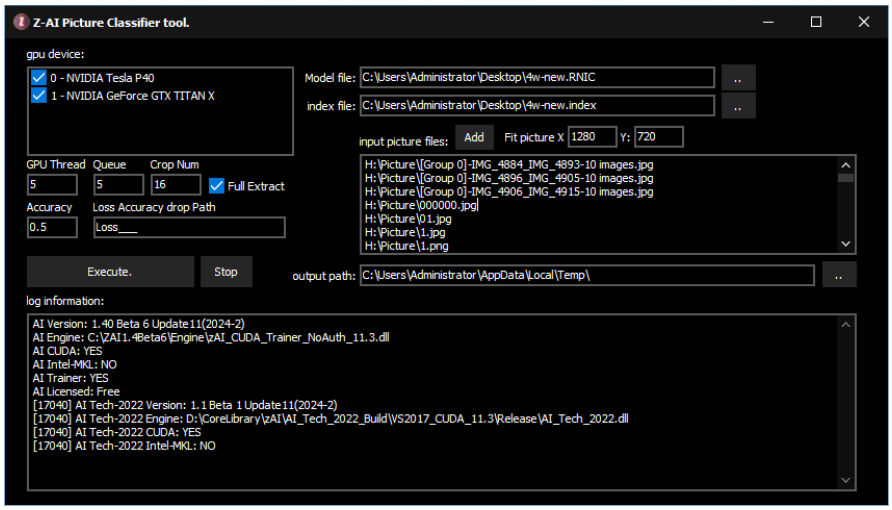
简而言之,巨程序是未来 AI 业务框架的进化目标,同时巨程序也会伴随着,自动化,大数据化,目前在 Z-AI 体系中的巨程序就是第六代监控体系.而 RealTime Tester/DNN-Face3.0/DNN-queue 这些框架更偏向简单流程.AI 业务级的巨程序概念也是在六代监控开发过程中被提出来并且应用的,目前 6 代监控巨程序是少年时期,少年不同于婴儿,少年是可以经历一些小风小雨摧残的,然后,在摧残中成长.

大数据处理

场景识别本质上也是一种图片识别,例如当图片数量繁多以后,需要对图片进行相似性采集,给图片做分类,这时候训练一个 ResNet 图片分类器模型,然后再批量化的运行一下识别程序,茫茫多的图片也就被自动化分好类了.这些分好类的图片也许并不会非常准确,但相比不分类会是一种极大程度的过滤效果,尤其是大数据.这种方式就是经典图片分类器.这种分类器在经历从提出到普及,再到沉淀至今,已经有非常多的应用场景,例如搜索引擎中的图片搜索,某些购物网站的以图搜图.

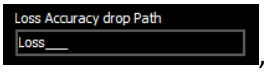
经典图片分类器工具通过下列工具菜单来启动,分类完成后,目标图片会归纳到不同的分类目录中,该工具是走多线程模型的标准 gpu 程序,也就是 DNN-Thread,当 GPU 越多, 分类执行效率也会更高.





当该选项被 

打开时,符合分类条件 0.5(50%相近度)的图片会在不同分类目录产生同一个副本,例如图片 1 既有猫也有狗,那么猫目录和狗目录都会有图片 1.



,这里的 Loss__表示当目标图片不符合匹配相近度时,所存放的子目录.经典分类器的问题是非常死板,并且不易于标注,当目标图片并没有被 ResNet 分类器的训练过,多半就会无法识别,这将会存放于 Loss__目录中.

,0.8 表示与目标分类具有 80%的相近度会执行分类,如果低于 0.8 则把图片放到 Loss__目录.

Z-AI 在 6 代监控体系建模场景模型时很暴力:先通过采集系统对不同的时间段进行采集,例如凌晨 6-7 点,深夜 8-11 点,这个时间段就作为夜间样本,采集出 500GB 的视频,然后用视频视频解码工具把视频解码成图片,大概有 1-2 万张图,不去仔细检查,直接暴力标签:夜晚,这时候,日间状态也是依法复制,最后,场景模型也就识别出监控目标的时间段了,而后面的分支,就是夜间和日间各自走不同的流程.

经典分类器模型需要数据覆盖,例如用场景 A 去识别场景 B,结果往往非常不准确.而通用分类器例如搜索引擎,购物网站,这类模型是依赖于大数据建模而来的,这种大数据会依赖于数据源+人工标注而来,经典分类器并不能天然具有分类效果,需要先走生产过程,然后它才能正常工作.

Barlow Twins: Self-Supervised Learning via Redundancy Reduction

这里简称 SSL,这里不做方法讨论,SSL 是一种使用向量语言的图片分类器,要对图片分类,需要在向量语言基础上做 K 值搜索,既临近 K.因为 SSL 使用向量语言,所以就不需要再对图片进行人工的标注了,只要暴力的给 SSL 喂图片,它会自行理解图片中的差异,理解结果是一种向量语言,并且这种向量具有线性机制,因此可以被计算出相近度.这对分类器的生产和建模来说,是一种革命性意义:当数据不再需要标注时,可以无节制的拿图给 GPU 学习,我们可以把 nnTB 的监控视频解码成图片海,然后让 GPU 学习,这些图片,最终,将被自动化的分好类.并且,这还不仅仅是分好类,这还可以辅助目标分类器,检测器的标注工作.给图片打标签是一种非常耗时的标注工作,并且这种标注工作是由人来干,会出现错误标注,试想一下:当标注工作被省略了,只需要给 gpu 学习无限的图片....

SSL 应用模型:不需要标签,自动给大批量图片分类

目前可以确定 SSL 能够在不用标签的条件下,使用巨量数据源训练,训练 SSL 会很费 GPU+时间,待训练完成后,可以自动的对目标图片进行分类,而分类结果会是形似和包含:图片与图片之间有形似点或者图片中的内容互相包含.当分类完成后,SSL 无法知道分类标签,因为分类标签未知,会导致未知分类繁多,也许分类的数目会远远高于训练所使用的图片数目.这时候,需要依赖于 UI 或脚本程序对繁多的未知标签进行程序化的管理,目前从已有的数据条件来看可以借助于快速搜索相似图片,可视化图与图的相似区,通过在 UI 中的一系列操作做到从茫茫多的未知分类定位出自己需要的分类,简单来说,SSL 的自动化分类会给出 1,2,3 个分类,这些是无标签分类,然后通过 UI 工具+一系列操作来实现需要定位的目标分类.在实际操作中的复杂性会比本文繁琐,未知分类也许会大达到数千个,定位到需要的目标分类会等同于挖掘到需要的数据.

用技术性语言来翻译 SSL 无标签的自动分类:SSL 的母体模型使用向量语言,当 SSL 母体针对 100 张图片进行分类以后会输出一张向量表.其中,每一张图片大约会有 100-1000 个局部向量(**每个向量=2048 个 Double,大约消耗 16K 内存**),这些局部向量用于表示图片中的各个区域,而这些区域都是随机的.从总体来看,100 张图片,从母体输出以后大约会有 10000-100000 个向量,这时候,**不可以使用统计学和机器学习的方法直接对这一大批向量做数学上的分类,例如 SVM 聚类法,MeanShift,拐肘法 K-Mean,因为这些方法是对向量做局部化的聚类**,也许一张图会被聚出 10 个类,而这 10 个类充满未知,这与 100 张图片发生关联并不在同一个层级上,不应该用数学方法往 SSL 的向量语言层路线前进,局部的前进只会折腾半天进步一小点,正确的方向是从 100 张图直接关联 100000 个向量的路线上走,反推会是一种不错的方式:先训练母体,然后反推 100 张图,让每张图都得到 1000 个局部向量,然后保存起来,待使用时再从向量库推导出线性化的结果,例如图片 1 有多少图可以与之匹配,以及匹配的相似区域数据,这时候,也就为自动化分类提供了可以展开工作的数据条件了,这会需要许多 UI 程序的交互操作.用最直接+简单的话来说,当 SSL 的母体模型出来以后,可以是点一下鼠标,目标图片群就被自动化的完成分类了,这时候 UI 的程序进入工作状态,对自动化分类进行交互操作.本小节是讲解 SSL 自动化分类的实现原理,方向,思路,技术方案的平衡化选择.

SSL 应用模型:场景分类

经典场景模型 ResNet 是走数据+标签训练,训练 SSL 只需要数据,而标签是在 SSL 母体被训练出来以后,通过反推训练样本的标签,得到向量+标签数据库,也就是.learn 模型.

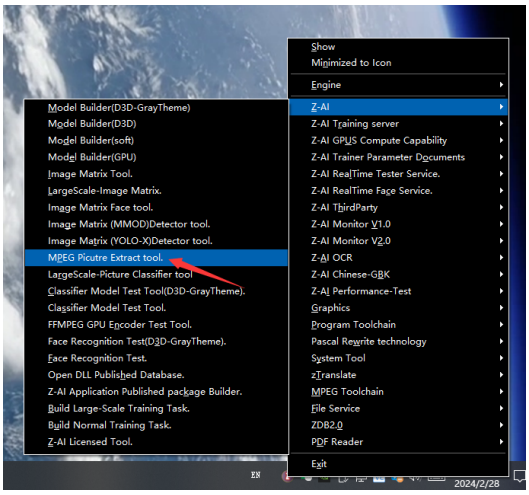
经典场景模型的工作是输入图片,直接返回分类标签识别结果,SSL 的工作是输入图片,返回 100 个向量,再用向量从.learn 模型进行查询,得到场景分类标签的识别结果.SSL 会优于经典场景模型,同时 SSL 的建模和使用也会比经典场景更为复杂.

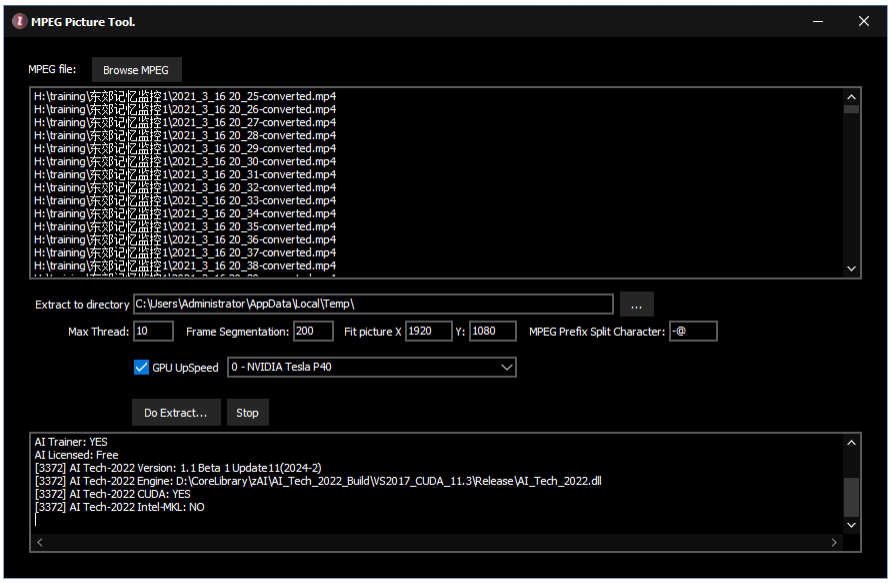
例如在火灾场景识别中,星星之火在用场景模型和 SSL 都会无法识别,只有大片整幅图的火灾才可以被场景模型识别.星星之火,例如在监控视频中点燃一张纸,这种识别需要使用检测器来干.火灾识别方案是场景模型+检测器共同行动的方案.

建模场景模型

Z-AI 体系建模场景模型大都会走视频路线,例如机场在视频数据中,需要有各个季节和各种时段的视频片段,季节的不同会影响行人穿衣的差异,时段的不同会影响光照,日间会是阳光反射+日光灯的机场环境,夜间会是荧光灯为主的机场环境,这类视频片段在监控系统中非常容易找到的,直接通过时段下载视频即可.

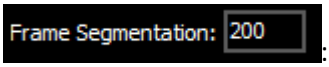
然后通过工具链菜单打开视频图片解码工具,这是使用 GPU 加速技术的视频解码转图片工具





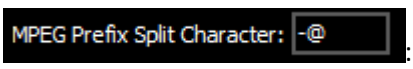
:开 10 个 GPU 解码线程,nvidia 的 gpu 会有一个解码器表单,gpu 在芯片中通常会内置 2-4 个硬件解码单元,当开 10 个解码线程,这些解码单元是按队列化对视频做逐帧解码,解码不会让 gpu 进入 100%的功耗状态,大多会处于 20%-40%之间,普通民用的游戏 gpu 例如 3090,4090,几乎没有解码会话的限制,解码会话是指同时使用 gpu 解码器的 api,例如使用 10 个 gpu 解码线程就会有 10 个会话,普通民用 gpu 最高解码会话大约是 50.解码线程并不会越高就越快,内部会受内置内置的解码单元限制. 当没有勾选 

时会使用 cpu 进行解码.



:解码时每间隔 200 帧提取出一张图片

:提取的图片会将强制拟合分辨率为 1080p,使用 ResNet 经典场景分类器会对图片分辨率有规格要求,通常会统一化的 720p/1080p 规格,因此在提取图片时也需要符合这种规格.SSL 场景分类器对分辨率没有要求,但会要求尺度比例,后面会详细介绍.



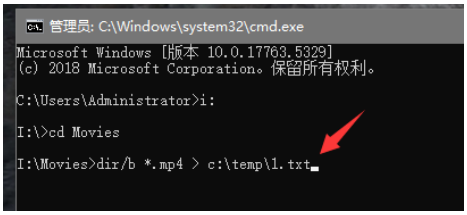
:以视频文件名-@符号的前缀作为裁剪的目标图片文件名

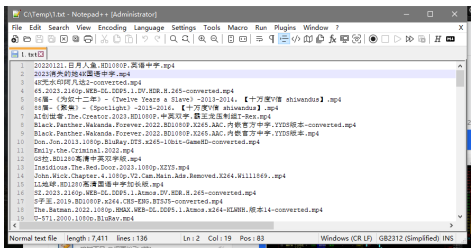
例如视频文件名是 **机场 A 候机厅 1@2024 年 3 月 1 日 10 点 30.mp4**,那么解码出的图片就是 **机场 A 候机厅 1.jpg**

在 Z-AI 的 6 代监控系统中,监控视频的采集文件名都为 **监控点@时间.mp4** 这类格式,从 6 代监控下载视频都是一次下载几百路监控,然后批量展开,这时,每一路监控视频的解码图片都会放在它所对应的目录中.然后直接把整个目录导入图矩工具就训练了.这些图片会数以万计,这是大数据的做法.

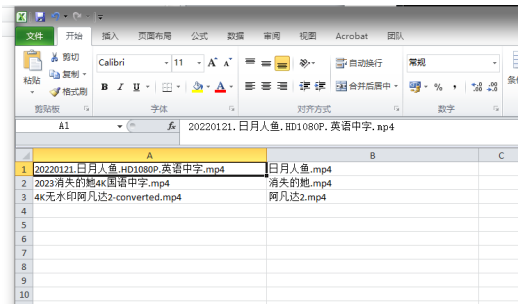
规范化视频命名的技巧

这里是以规范化视频命名讲解文件命名的技巧,在大数据处理中,大规模文件改名用的非常多.这里以我日常下载电影视频为例,先通过命令行工具,使用 dir/b *.mp4 > c:\temp\1.txt,这是把目录中的所有 mp4 视频文件列表输出到一个文本文件中,这时候可以打开查看和修改 c:\temp\1.txt,但是,先不忙在这里改

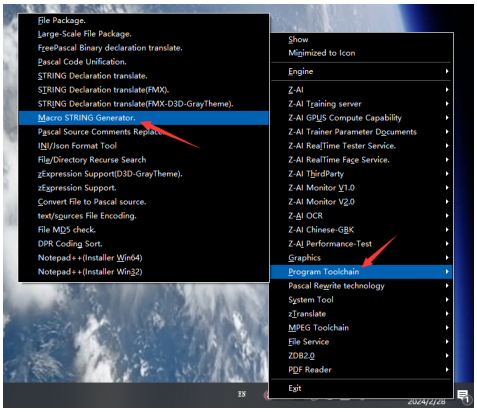


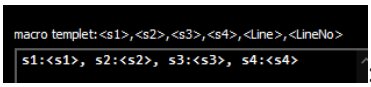


然后打开 excel,把这些文件名粘贴过去,A 是原文件名,B 是目标文件名,改文件名就修改 B 列中的名字,在 excel 里面在 B 列里面加点序列号,格式时间,都是可以的,这一步工作是编辑文件名的数据,相当于用 excel 生产数据.

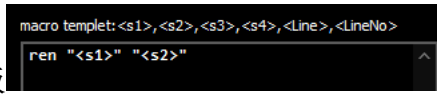


然后通过工具链菜单打开宏字符串工具,然后把 excel 的数据粘贴到中央框框中,excel 的表格分隔符默认是#9





:这个时候开始修改左边的宏模板



<s1>,<s2>表示第一列和第二列,Windows 修改文件命的命令是 ren “源文件名字” “目标文件名”,写完点 build

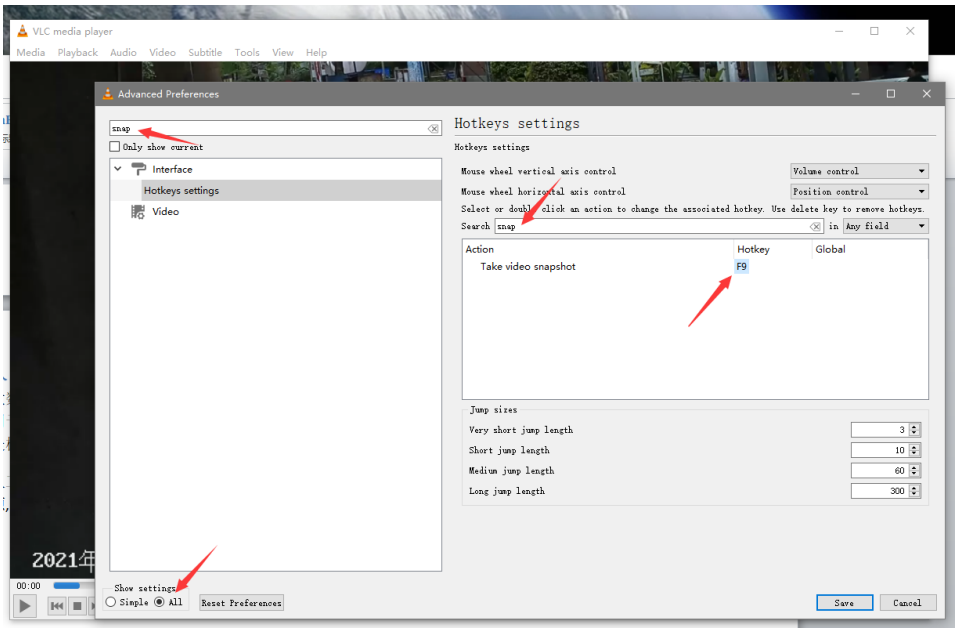
,这时候文件批量化的脚本类命名也就出来了.在准备场景状态视频时,可以通过这种方式批量的进行分类,例如范冰冰的一批视频就弄成 **范冰冰@123.mp4** 这样,就可以把范冰冰视频的图片全部都生成到 **范冰冰** 目录中了.编写程序跑去弄这些事情效率太低了.

人工方式从视频提取图片

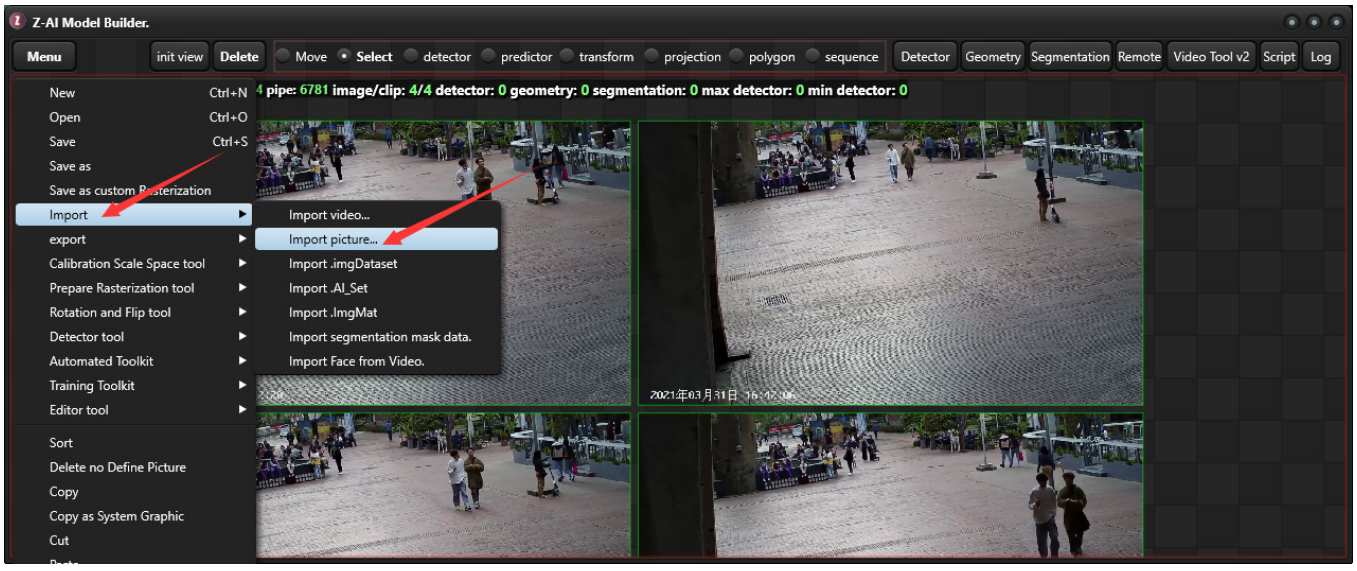
大数据做法是直接用工具从视频提取图片,这些图片数量非常庞大,因此大数据工具所提取的图片一般都可以直接用于训练场景模型这也是人们最喜欢的一种建模方式,简单,自动化,高效率,人们的时间被解放出来,大部分工作都是机器在干.

人工提取视频图片,是用 vlc 结合标注工具,亦或是直接用标注工具来导视频图片.如果使用 VLC 需要先准备好视频源,然后在 VLC 播放,快进,通过 VLC 热键截图,然后再导入到标注工具.

使用 VLC 首先切换语言为英文,打开参数设置,切换到 all 参数方式,然后搜 snap,如下图,最后把截图热键改成 F9

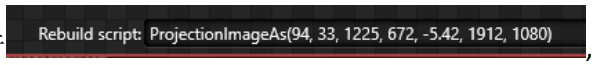


Vlc 工具的优点是可以针对视频做快速定位,F9 截图会保存在”我的图片”文件夹,直接导入到标注工具

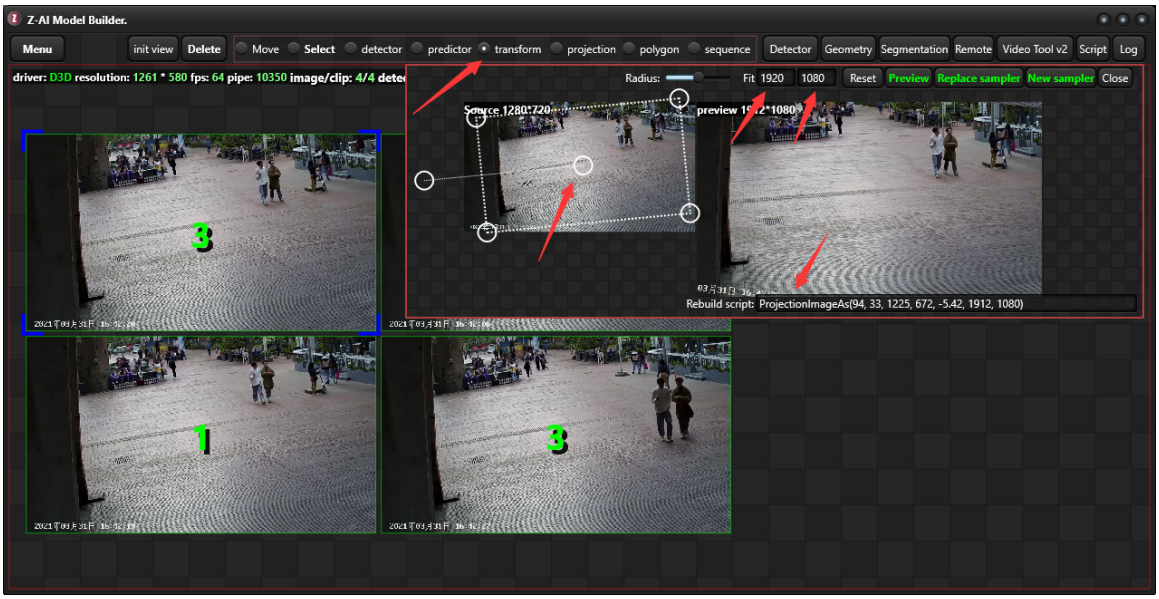


批量的矫正 vlc 的视频截图

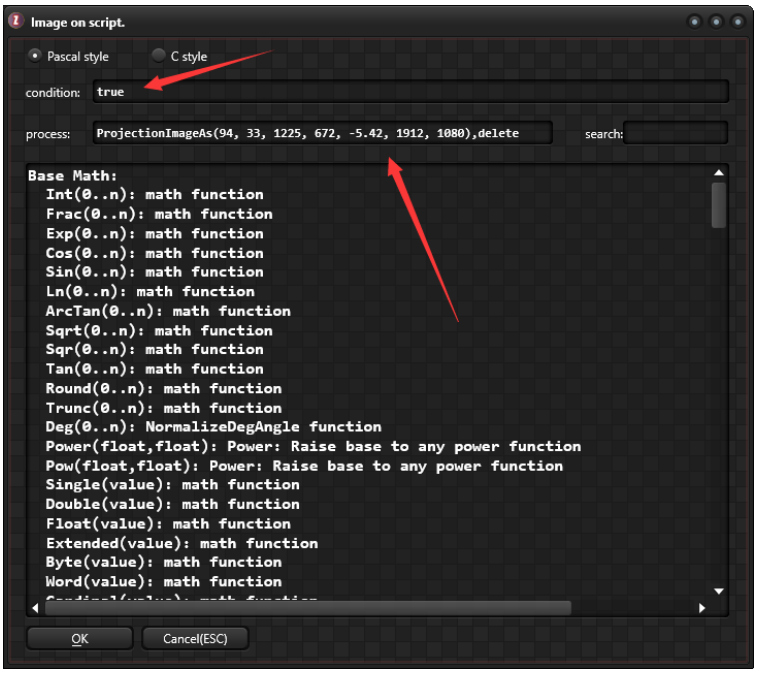
视频源大都会是 720p/1080p 这类规格,如果视频源不符合规格,截图也会不符合,亦或是视频源的截图歪歪斜斜.

操作方法解释:切换到 transform 工具,尽可能大的框住图片内容,如果视频歪斜,可以拖动一下框框变形,让它最终输出一个比较贴近效果,这时候,注意下面这行 

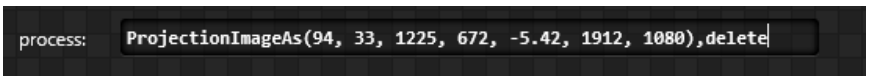
,这是变换内参实时转换出的一种脚本,这可以把单张图的变换直接全局化,也就是从 vlc 里面采集出的所有图片,一律按指定方式进行变换.



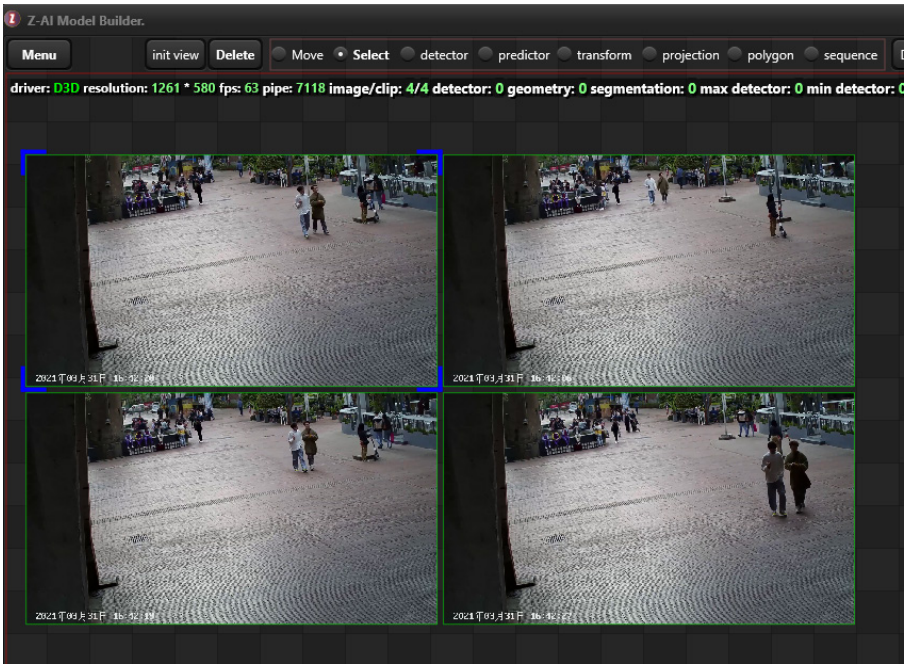
然后点开脚本,把变换内参代码复制进来执行,在内参代码后面加一段”,delete”



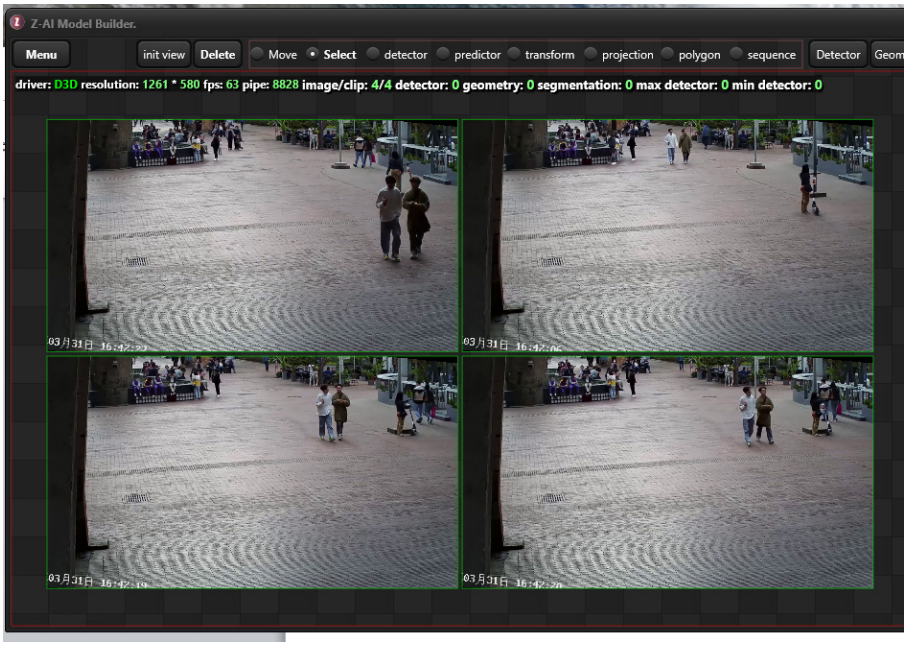
后面加一段,delete,表示执行投影变换以后,把原图片删除掉



使用脚本变换执行前,图片规格为 720p,监控画面轻微歪斜

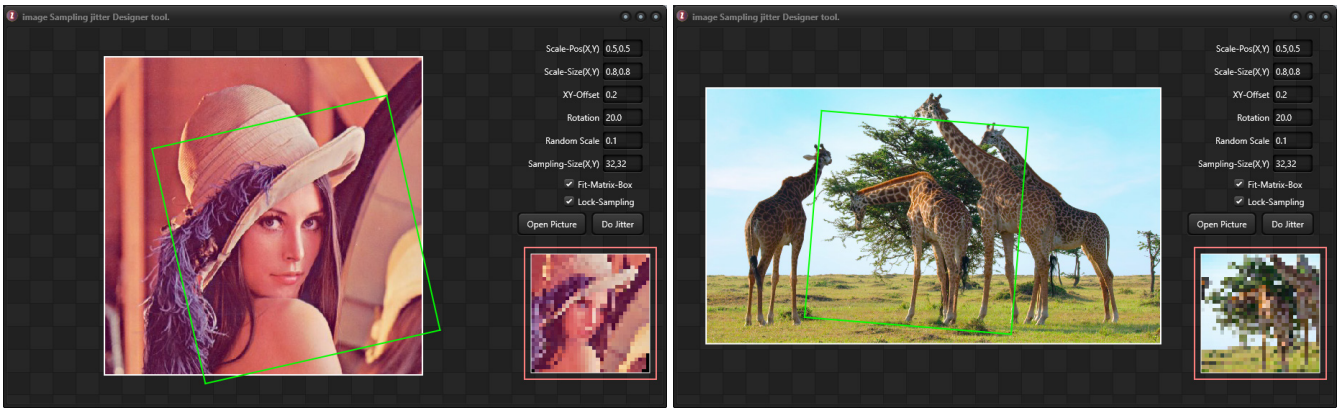


执行变换后,图片规格为 1080p,歪斜效果被矫正.

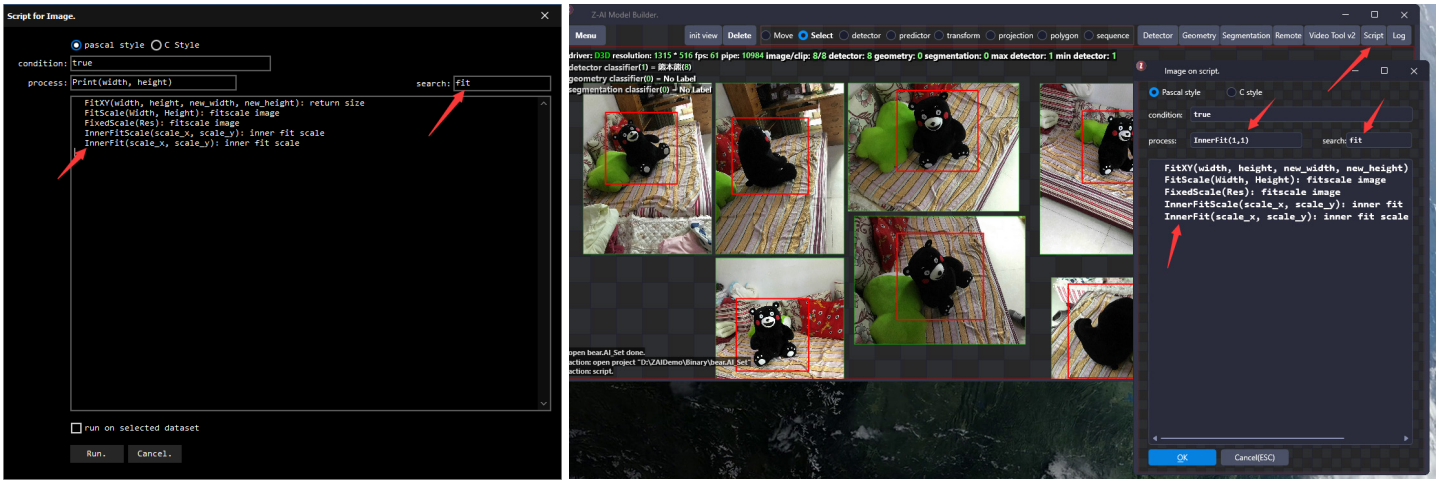


对 SSL 模型使用的图片做尺度矫正

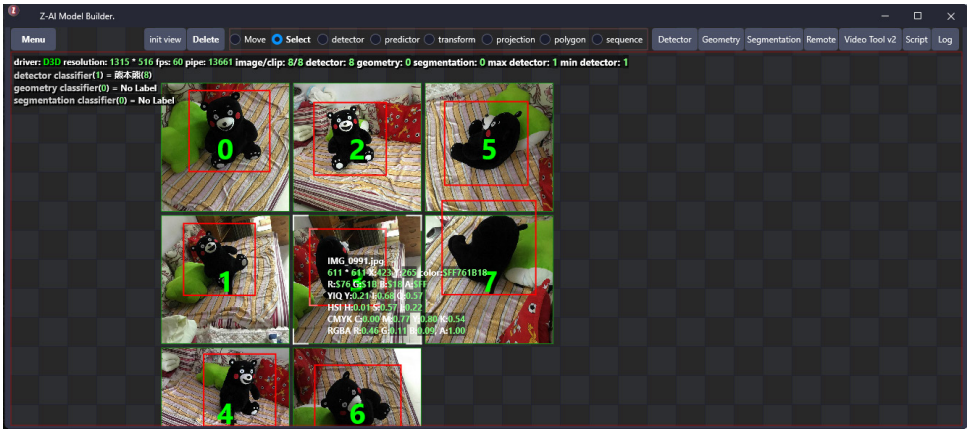
下图是一个 SSL 建模时的抖动参数设计工具,用于预览抖动效果,当确定绿色框框可以有效覆盖全图以后,然后把抖动参数写进 SSL 的训练脚本中.抖动算法效果会被图片尺度所影响,例如左边是 1:1,右边是 16:9,这时候,绿色框框更偏向在 16:9 的中央跳来跳去,这在 SSL 训练模型时流程会观察不到 16:9 的两侧内容.



在多数时,图片尺寸都是不规范的,一个样本库会有 16:10,16:9,4:3 多种尺度比, 对此样本工具体系提供了脚本方式的矫正方法,不管是标注工具还是图矩工具,它们都可以支持脚本批处理,下图以图矩的脚本为例,搜索 fit(拟合),在应用模型中,图片的 fit 都是 2d 方式,需要区分外拟合,这是图片以不走样尺度朝向目标分辨率拟合),内拟合,这是以目标尺度朝向最佳的图片内部进行裁剪,如果要把 16:9 这种图,以不走样 1:1 的方式剪裁出来需要走内拟合路线,这里给出的脚本就是 InnerFit(1,1)

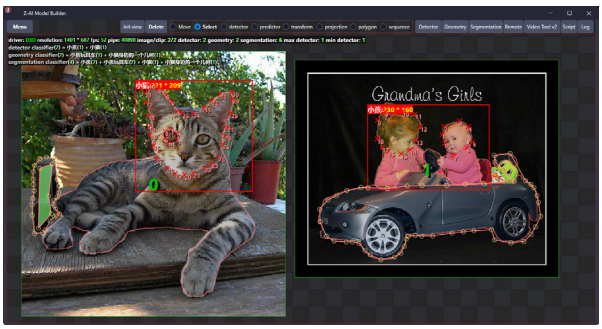
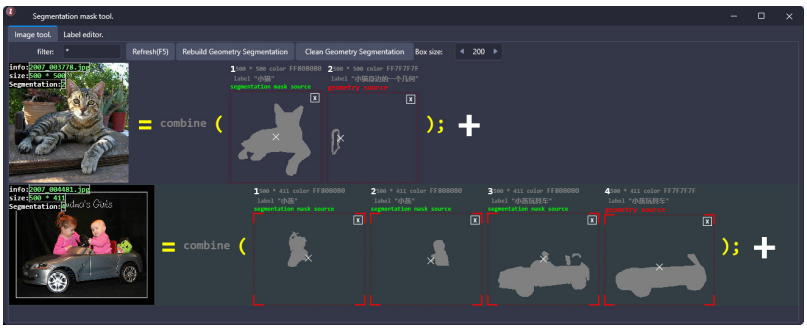


innerfit(1,1)是从图片尺度入手计算最佳的 1:1 目标剪裁尺度

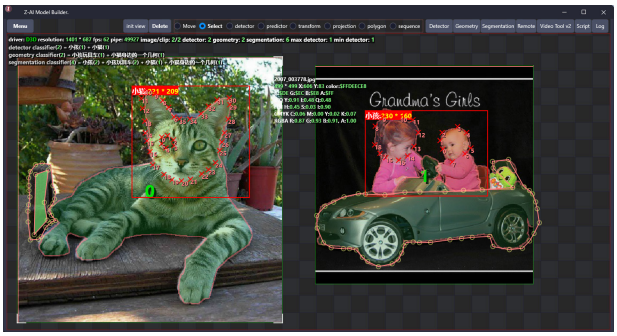
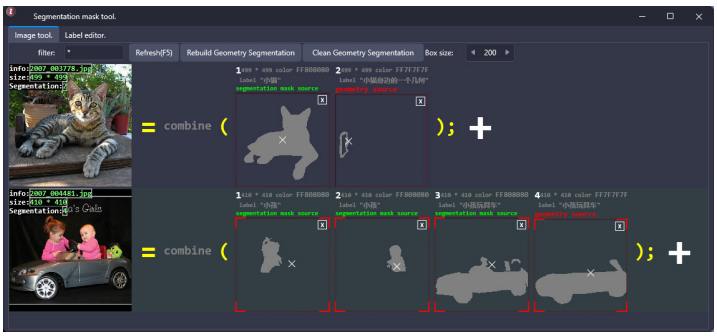


在样本工具链体系中,fit 类脚本具有数据重构能力,例如检测器标注,目标分类器标注,顶点预测器,几何分割器,像素分割器,在一张图片中往往会包含非常多的结构化数据,脚本中的 fit 函数被执行时,这些数据会进行 rebuild 形式的再计算,这是非常复杂的流程.

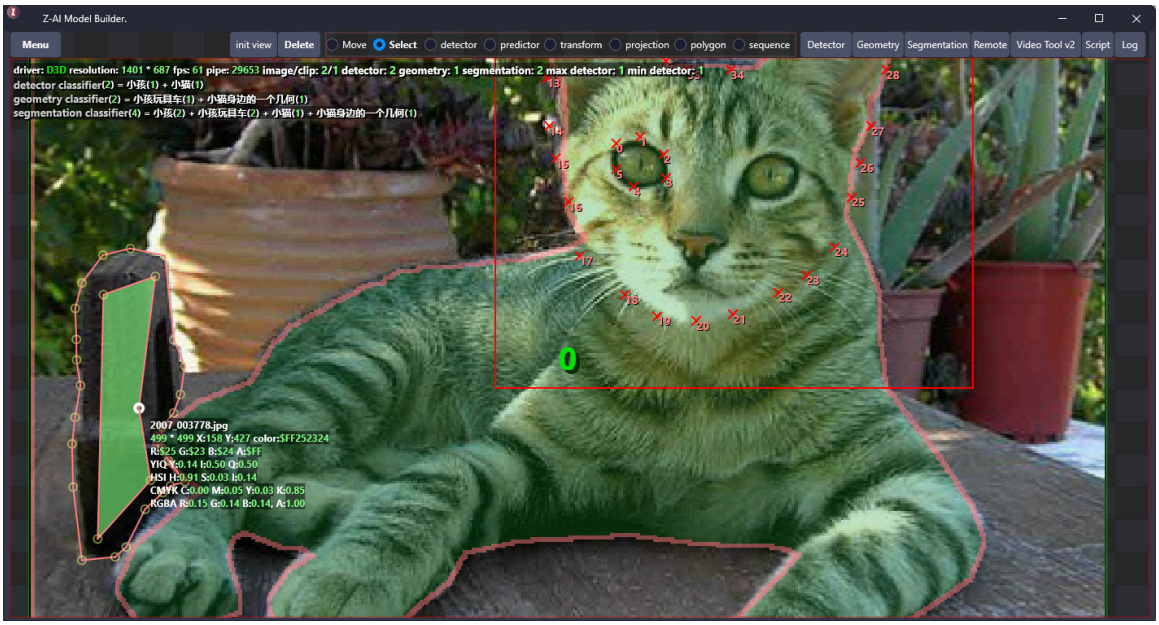
下图是一个 demo 样本库,数据基本齐全,在 Z-AI 的 Demo 目录中可以找到它

这时候,开始使用 innerfit(1,1),对这两图进行尺度矫正,数据被重构

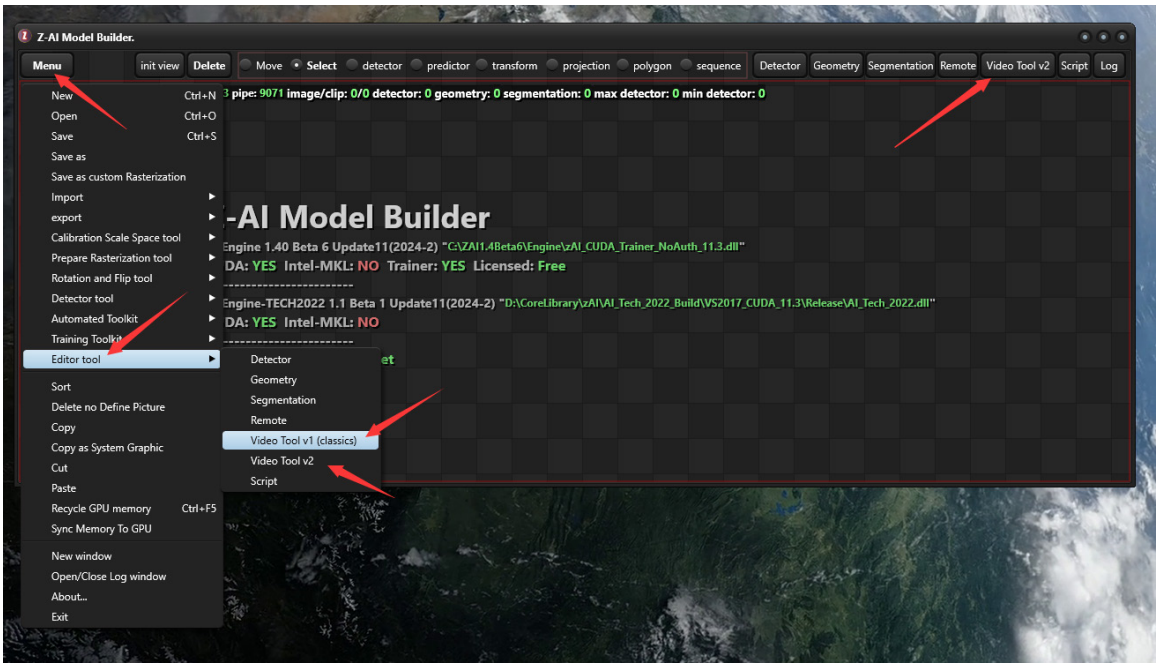
使用 Innerfit(1,1)重构后,几何标注并没有走样



总结一下本小节:**使用 SSL 建模需要结合抖动参数,矫正一下样本的尺度,使用 innerfit 函数不会导致标注数据的丢失.**

使用 Model Builder 内置的视频工具提取图片

Model builder 工具是走堆砌路线的局部项目,堆砌路线的含义就是如果要对某项功能做出推翻级的大更,那么被推翻的功能就会经历最后一次 debug 确定可用,然后保留下来,同时新的大更将取代老的.内置视频工具有 2 个版本,分别是经典版本 V1 和 V2.两个内置视频工具都可以用于提取视频图片.



下面以 V2 为例,点开 V2,指定视频文件路径(多文件以|符号切分),也可以是 rtsp/rtmp/http/https 这类流式视频 URL

Split:v2 是解码完整视频,有些视频非常长,例如 3 小时长度,这时解码会非常耗时+耗内存,Split 是均化切分,可以把 3 小时视频切分成多个 5 分钟的封装码流片段.这时再来用解码就会很容易找到需要的目标片段.

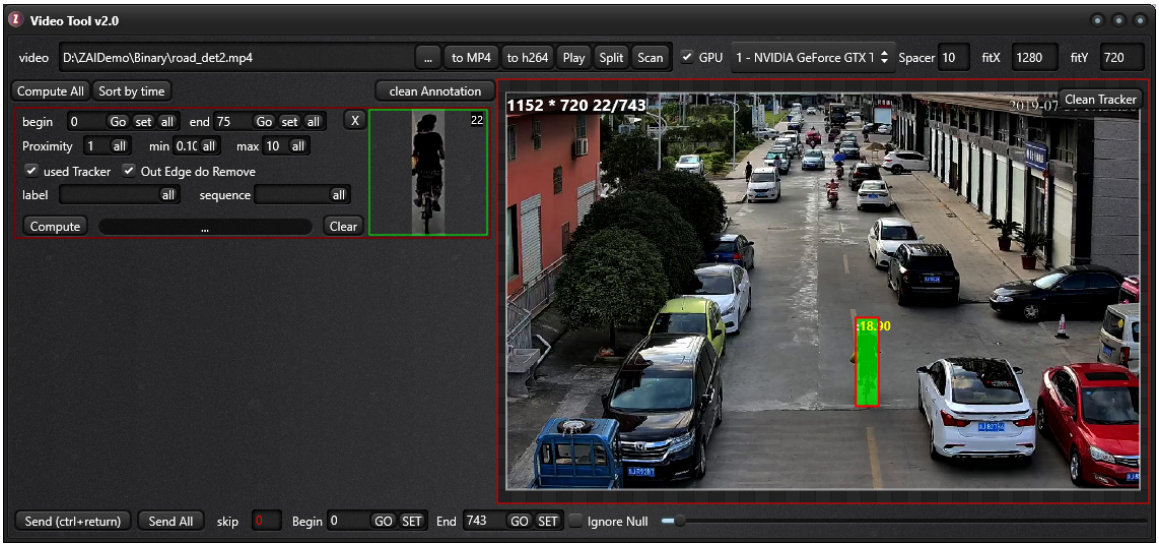
Scan:启动采集流程,采集流程的参数都集中在上面这排,采集过程可以随时中断,大视频采集会非常消耗内存

GPU 0 - NVIDIA Tesla P40:v2 可以选择指定的 gpu 对视频进行硬件解码,gpu 对于高清+高帧率视频的提速非常明显.

Spacer 10:采集图片的间隔帧,对场景模型来说,如果视频编码帧是 25fps,可以给 100,表示每 5 秒采集一张图

fitX 1280 fitY 720:拟合分辨率,v2 这里的缩放图片是走的硬件加速,这里的拟合分辨率会比脚本中的 fit 快 10 倍以上,如果 fitX+fitY 都给 0 会以视频的原始分辨率进行采集.

当采集完成后,如果在图片中拉框左边会出现 tracker 工具,这会锁定和跟踪框住目标,tracker 工具主要用于连续性的目标行为采集,本文以场景模型建模为主,tracker 工具环节直接省略.



Send (ctrl+return):v2 的运作思路是先把视频解码以序列帧方式暂存于内存,这些视频可以被逐帧浏览,当需要采集某一帧视频时就 send 一下,这时,会发送至标注工具然后排序.send 有检查重复检查能力,同帧视频连续 send 无效.

Send All skip 0 Begin 0 GO SET End 743 GO SET Ignore Null:Send All 是按 begin 帧到 end 帧循环的 send,其中,skip 表示循环中的跳帧,复选框 ignore null 表示忽略空标注,这些标注是行为采集的框框,例如从 0-100 帧有行为的 tracker 框,那么 send all 时就只会 send 0-100 帧,后面的帧将会忽略掉.

GO SET:go 表示当前进度立即转向到目标帧,set 表示将当前进度赋值,在 v2 工具有许多表示帧定位的数字输入框框都有 go+set,其中 set 就是把当前进度赋值于数字框框.

:进度条被鼠标点击以后会定位焦点,在焦点中可以使用键盘方向键配合 ctrl+return 采集.

当采集完成后在标注工具按下 ctrl 键会现实绿色数字,表示图片索引号,这不是帧索引,而是图片在标注工具的索引,标注工具的排序会以直方尺度进行,优先排序大尺度,如果尺度相等则会看到直方从左上到右下曾斜率下降排列,因为二维空间排序不是按索引进行.**当鼠标处于图片中,箭头所指的地方会出现图片信息**,该信息的格式为,**视频文件名_分辨率.帧索引**.只有这条信息才会是准确的帧信息.

场景模型数据规格

RNIC/LRNIC

全称 Deep Residual Learning for Image Recognition,

作者 Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

Paper https://arxiv.org/abs/1512.03385

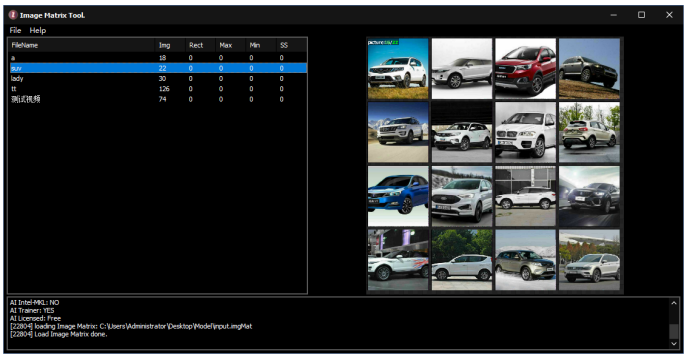
RNIC 是 2015 年 ILSVRC 获奖技术方案(NO.1),RNIC 在数据输入层面会走标签+抠图路线,这种方式被大量项目使用,并且延续至今.今天的 ImageNet 有许多变种,有些版本会把网络建到 1000 层+上亿参数,也有些版本会结合抠图+打框进行复杂标注训练,因为普及所以变种版本多.

RNIC 的额定分类 1000,也就是最多能识别 1000 种场景.LRNIC 的额定分类 10000,也就是最多能识别 10000 种场景.

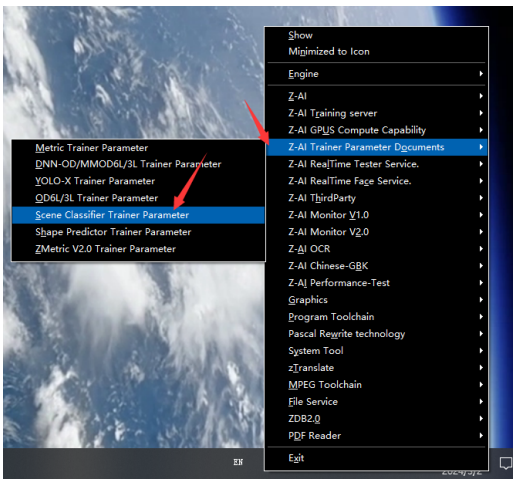
RNIC/LRNIC 要求图片分辨率至少>720p

RNIC/LRNIC 的数据样本模型设计它的定位是要求必须使用图矩(Image Matrix tool),在一个单独标签中可以有多张图,如果偏离设计之初使用.AI_Set/.ImgDataset 建模,那么将以样本中的标注框来替代图片.

RNIC/LRNIC 的标准样本集如下图,图矩是场景模型建模准工具,Model Builder 更偏向拉框标注,不适合场景模型



RNIC/LRNIC 模型训练参数文档使用工具链菜单打开



SSL

全称 Barlow Twins: Self-Supervised Learning via Redundancy Reduction

作者 Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, Stéphane Deny

Paper https://arxiv.org/abs/2103.03230

SSL 是一种无监督的图片分类器模型,paper 作者含有 Yann LeCun,可以当成顶会获奖算法来对待.SSL 对标签没有要求,只需要把图片暴力的仍进去就能完成分类器建模.而 SSL 怎么去应用,例如怎么做数据挖掘,怎么做场景分类,这由 SSL 使用者决定. 大家如果直接找 SSL 的文档或则 demo 这会毫无用处,SSL 是一种底层的数学模型,这是一种开拓型的方案,为 CV 圈开拓视野的方案,需要在 SSL 基础上自己做出二级模型建模方案这样 SSL 才可以有应用空间.

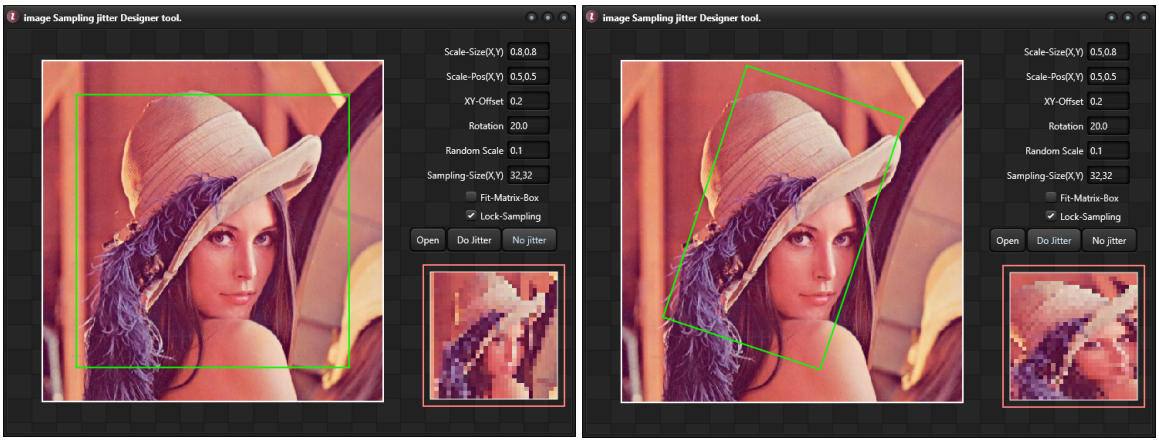
Z-AI 体系将 SSL 定义母体模型,而场景模型是在 SSL 母体基础上构建而出的二级模型,SSL 在母体训练完成后,只需要给出一个参照样本库例如,.ImgMat(图矩),SSL 会使用母体实时的对.ImgMat 生成向量库,在场景分类时可以直接从向量库反查出相似样本,这就已经包含了目标分类结果了,并且还可以有许多候选结果. 这与 RNIC/LRNIC 训练方式不同,而这种模型只是 SSL 在应用层的一方面.

SSL 母体的额定输出规格:单次 jitter 的向量长度为 2048 个 double,这些 double 以 learn 引擎进行管理,learn 对于 double 的局部加速计算有许多优化措施,主要使用指令集优化(sse)+候选算法优化(ca)+运行时优化(fast mode).

SSL 每次在输入网络计算时,每张小图的输入尺度可以动态的,但必须可以整除 8,默认小图尺寸为 32,每个步数计算的长度为 64 张小图,64*(32*32),训练约耗费 10G 显存.在大显存 4 卡或则 8 卡 gpu 平台,可以把小图尺寸调大.在单卡 P40 训练,速度大约每秒计算 150 张小图.

SSL 训练可以多卡提速:SSL 不会出现 DNN-OD/YOLO-X 这类单卡快于多卡,SSL 使用 2 张 gpu 合算提速约等于 2 倍.

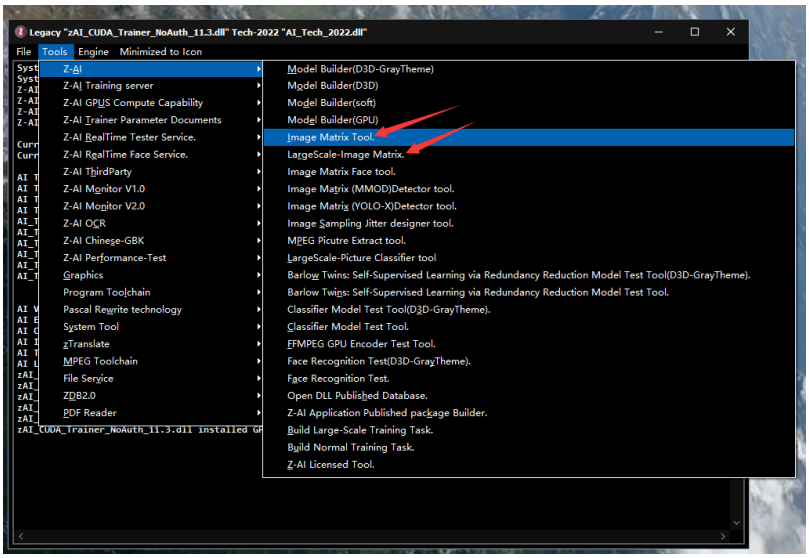
SSL 的小图生成机制为 jitter 机制,该机制为 pascal 独有:通过工具链来设计 jitter 内参,然后,SSL 训练母体时会使用这些内参来生成小图,例如下图,内参可以是等比,也可以非等比,这样来训练母体模型,既能全局识别(场景模型)也能局部识别(大数据挖掘,检测器挖掘).



SSL 的 Jitter 机制为:首次 jitter 不做抖动,随机抖动只会从第二次开始,例如在使用 SSL 做场景识别时,如果 jitter 参数给 0 or 1,那么就是直接全图识别(直接以无抖动方式设抠图).

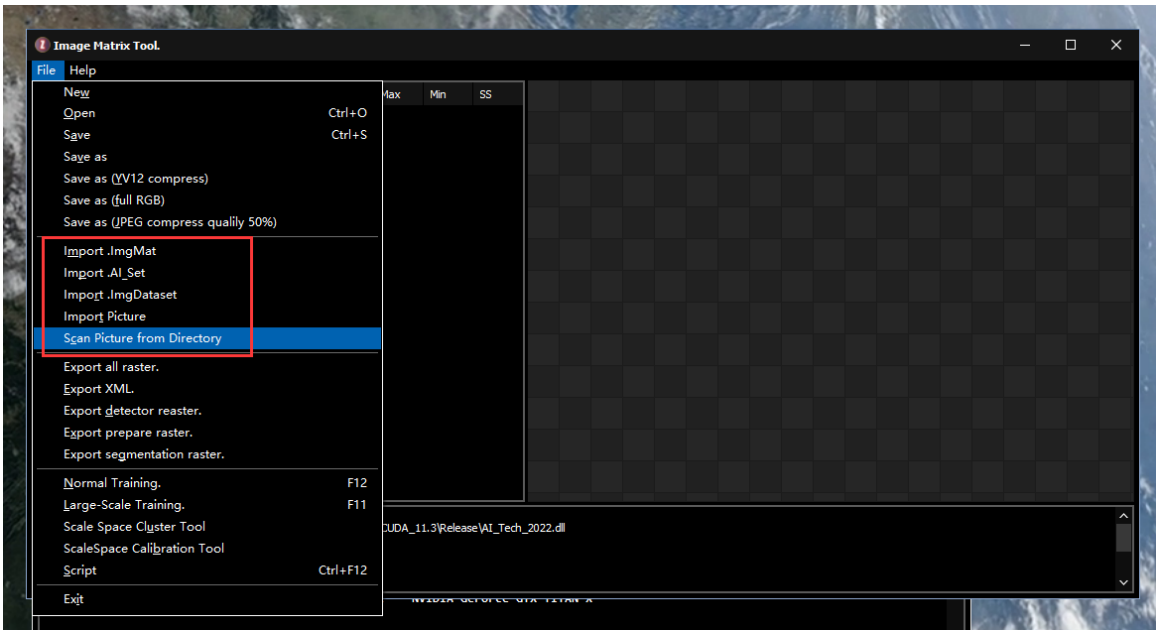
RNIC/LRNIC 建模

首先通过工具链菜单打开图矩工具,Z-AI 体系图矩工具有两套,以 Image Matrix Tool 为主,LargeScale-Image Matrix 是低内存模式的图矩工具,用于解决建模大数据内存不够的问题.

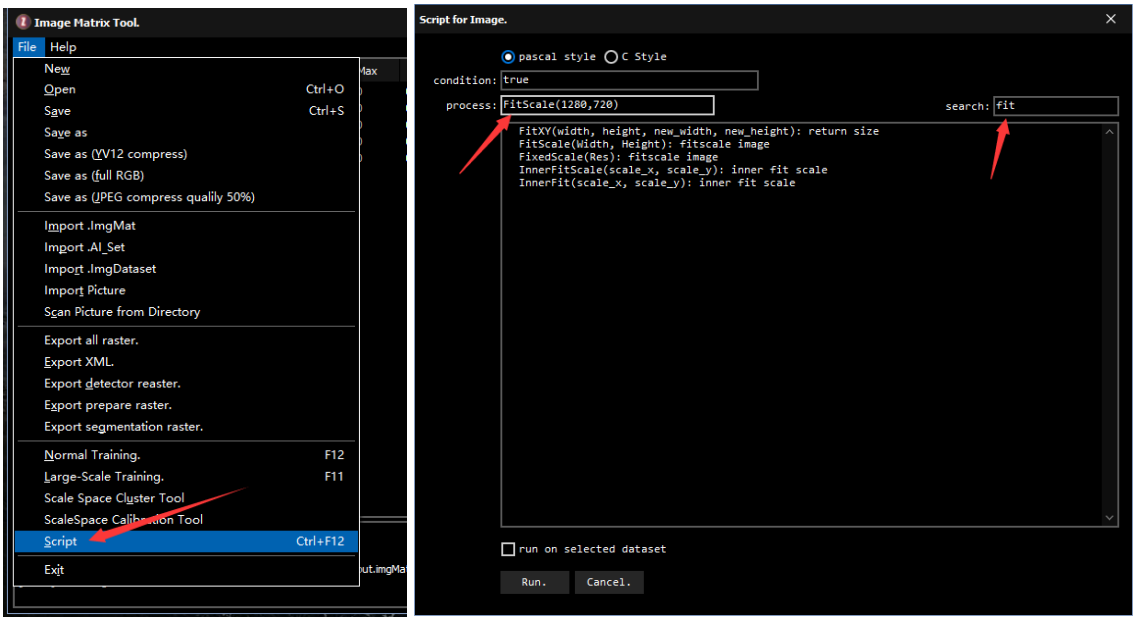


进入图矩以后,红框菜单都是往图矩里面导数据的功能,可以是各类样本库格式,也可以直接导目录.

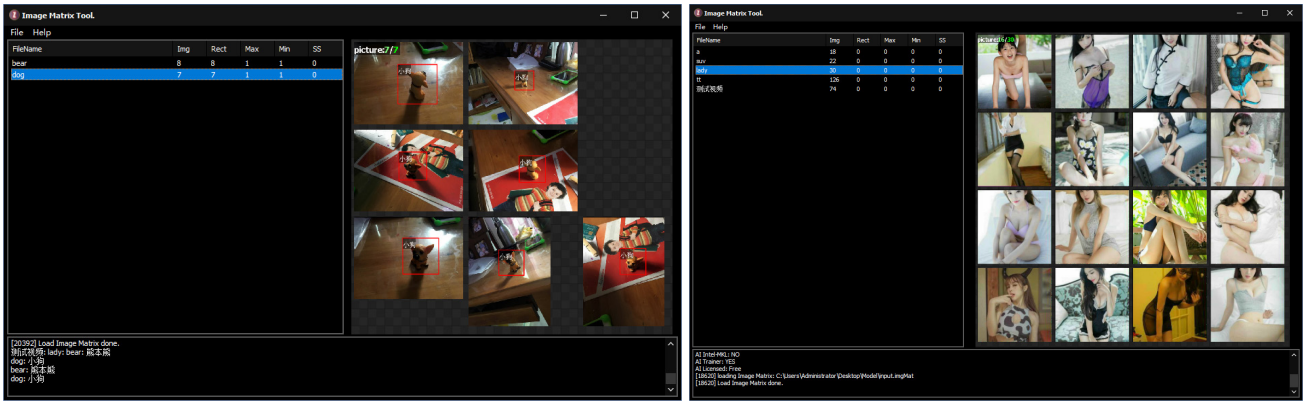
通常来说**做场景分类样本库就是直接导目录**,先把需要分类的标签以目录命名,然后再把各个图片放到目录中,最后,直接在图矩里面把整个目录导入进来,当然,也可以是走局部数据路线:针对单独分类单独建.AI_Set/.ImgDataset 样本库,参考,[人工方式从视频提取图片](#)章节



尺度统一化通过脚本来干,点开脚本,搜 fit,使用 fitscale(1280,720),把图片尺度全部拟合成 720p



图矩的默认预览规格为 4*4,每次点选一个类别,预览会先随机抽取 16 张图,然后全部拟合成 300*300 分辨率,最后,以 4*4 形式画出来



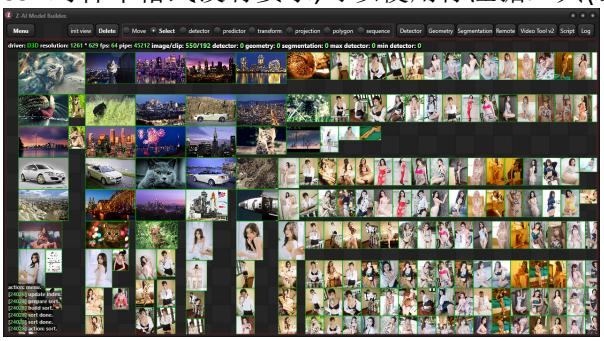
当确定分类标签无误以后,就可以开始训练了,RNIC/LRNIC 支持多 GPU 训练.

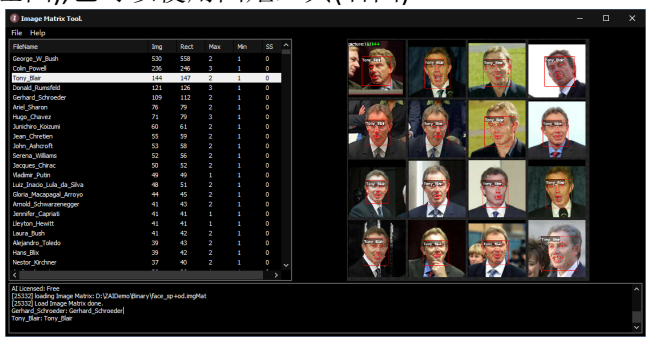
因为图矩都是大数据路线,因此训练需要区分,Normal 训练(直接放内存训练),Large-Scale(放硬盘中训练),具体细节可以参考超算服务器使用指南的相关文档.

10 万张图的场景样本,通常 1 在天即可完成训练,除非发生 loss 无法迭代下降,这种情况是错误标签导致的,两张完全相同的图,被打了多个标签,RNIC/LRNIC 找不到差异,直接强制结束就,进入测试环节,只要测试通过,后面都按正常方式使用.

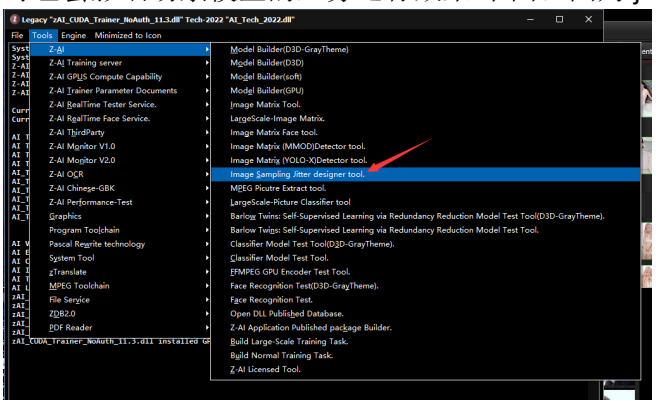
SSL 建模

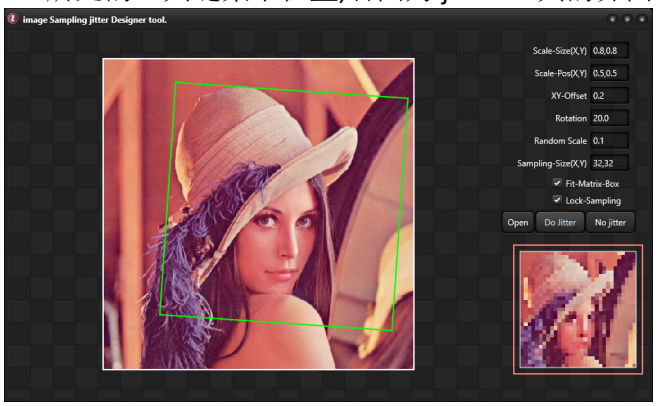
SSL 对样本格式没有要求,可以使用标注据工具(左图),也可以使用图矩工具(右图)





SSL 建模关键在于 jitter 内参值,这些值决定了 jitter 如何从图片中抠图,这会直接影响母体对整张图的识别效果,同时也会影响场景模型的业务运行效果.下面左图为 jitter 所处的工具链菜单位置,右图为 jitter 工具的界面





原始框参数,jitter 需要原始框框作为计算参照,No Jitter 按钮是无抖动直接画出原始框

Scale-Size(X,Y) 0.8,0.8 :抖动框尺度,该尺度以最小边长计算,计算公式,Width=(min edge*0.8),Height=(min edge*0.8)

Scale-Pos(X,Y) 0.5,0.5 :尺度坐标值,计算公式为,X=(width*0.5),Y=(Height*0.5)

下列为抖动参数,在原始框生成以后,以原始框为参照进行抖动计算

XY-Offset 0.2 :坐标抖动,计算公式,X=(Box X+Box Width*Random(-0.2,0.2)), Y=(Box Y+Box Height*Random(-0.2,0.2))

Rotation 20.0 :旋转抖动,计算公式,Angle=Random(-20.0,20.0)

Random Scale 0.1 :缩放抖动,计算公式,Scale=1.0+Random(-0.1,0.1)

小图生成参数

Sampling-Size(X,Y) 32,32 :生成规格为 32*32 的小图,SSL 的默认小图规格 32*32,SSL 对小图的观察模式为 RGB 彩色.

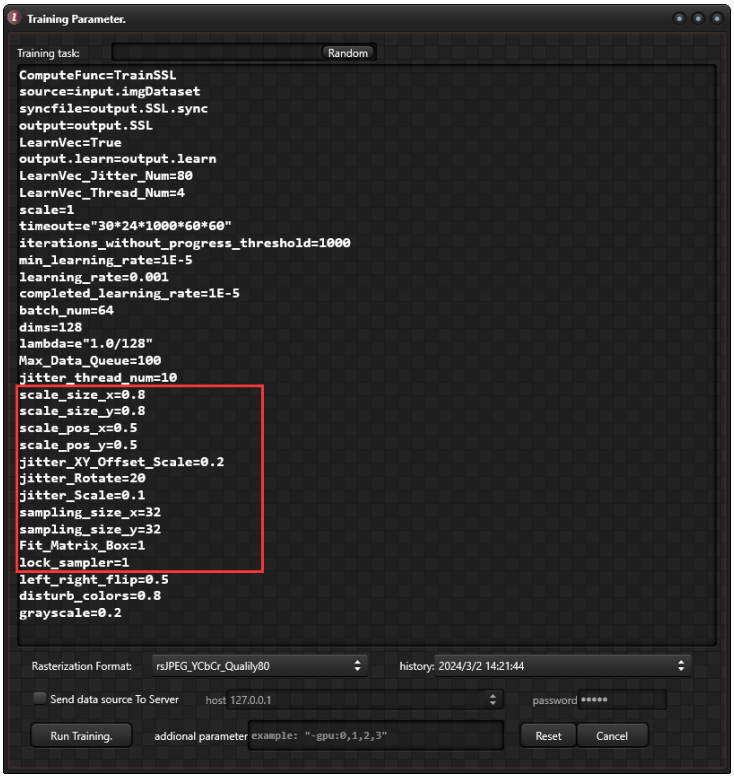
☑ Fit-Matrix-Box :如果抖动框生成的尺度与 32*32 的小图规格不符,那么将使用最小走样机制对框框尺度做拟合处理

☑ Lock-Sampling :如果抖动框坐标越界以黑色填充采样

在设计 jitter 内参中,通过 Do Jitter 按钮让框框动态抖动实时观察抖动的范围是不是可以捕获到图片内容.

Open :Open 按钮可以打开图片,通常可以选择一张会被训练的图片来测试内参,也可以选择打开 SSL 模型,这时候,内参值会以 SSL 模型值为主.

当 jitter 内参确定无误,这时候,在 SSL 训练参数中将对应的值填上即可开始训练.



SSL 训练完成的模型会自带 jitter 内参,这些内参在 SSL 运行中会直接影响对目标的采样规格,这样也要求在训练模型给内参的步骤必须考虑使用场景,例如尺度给 0.8*0.8 那么就是全图识别,SSL 模型在运行中也会使用 0.8*0.8 抠图进行识别.

SSL 的训练完全可以使用多 gpu 方式来干,这会明显提速数倍,对于大规模图片的场景建模来说,多 GPU 训练 SSL 是必须使用的方案.

SSL 训练中,Loss 并不会像 RNIC/LRNIC 那样走的非常低,一般来说 Loss 可以低于 5.0 那么就可以直接使用了.

已发现的某些谜之问题:当尺度给 0.8*0.8 用单 gpu 训练,rxpci 每秒消耗 6-8GB 输入带宽,而用多卡 rxcpi 则只消耗 100M 输入带宽.

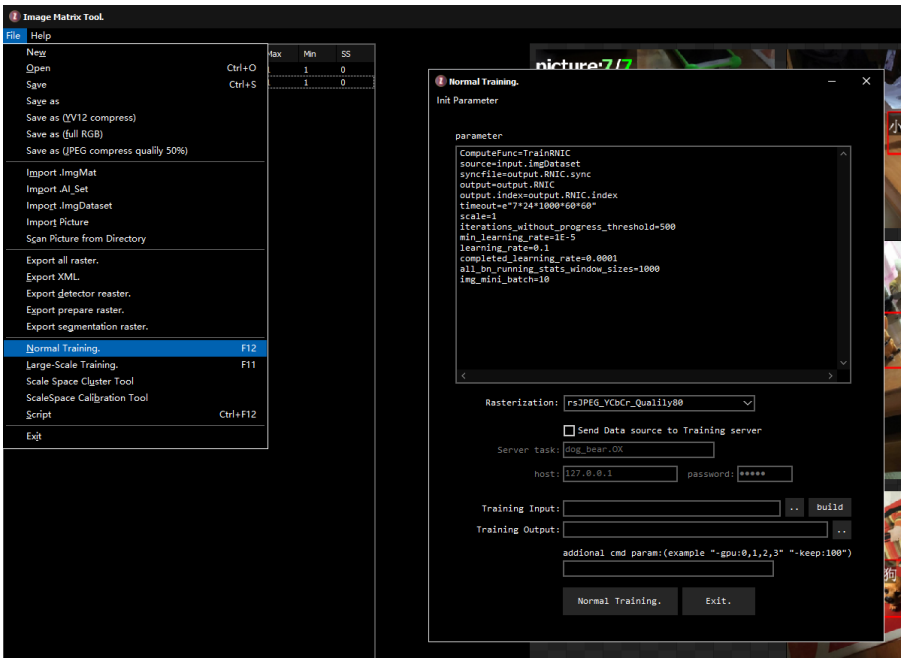
SSL 在跑识别流程时,内置缓冲区一次可以处理 batch_num(64)张小图,例如 SSL 识别一张图,抖动 100 次,那么会先计算前 64 张小图,计算完 64 以后,再计算后面的 36 张,即使一次抖动 1000 张小图给 SSL,内部缓冲区也会是按这种机制工作,不会一次性把小图全部放到 gpu 显存.

在另一方面,ZM2/ZM1/Metric/LMetric 是一次一张小图,通过频繁的调用 api 做到目标识别,SSL 从底层设计是单次多小图识别模式.因此 SSL 在业务运行中对 GPU(DNN-Thread)+CPU(Learn)算力消耗会高于 ZM2/ZM1/Metric/LMetric 数十倍.

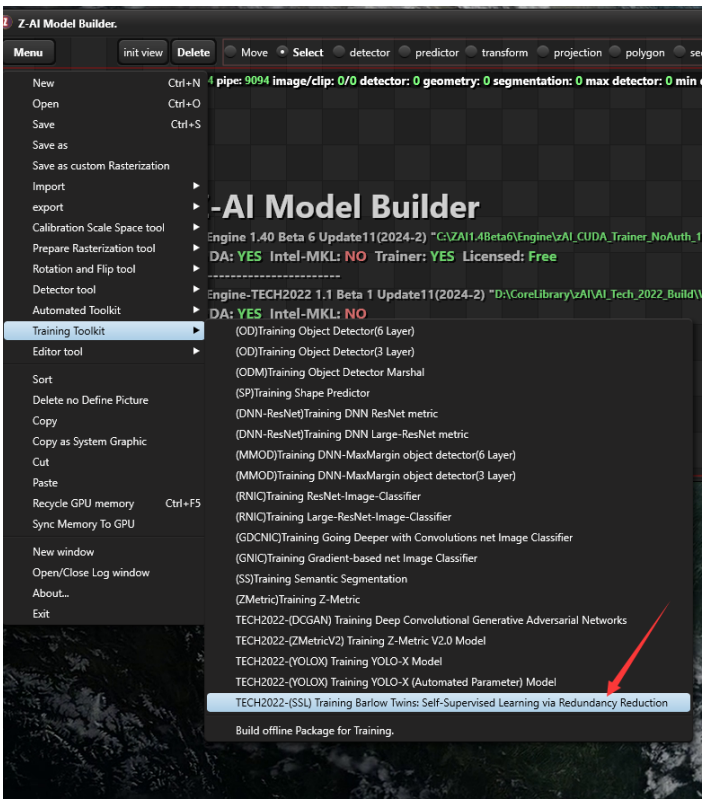
总结:SSL 因为不需要标签的特性让它可以直接处理大规模图片数据,SSL 建模的关键在于抖动框内参.SSL 有许多可以开发的技术点,目前 SSL 只是被使用到了场景模型中,未来 SSL 还能作为大数据挖掘的重要技术方案.

训练场景模型

RNIC/LRNIC 必须使用图矩训练,切勿在标注直接训练!另一方面,RNIC/LRNIC 样本库通常会很大,因此 Large-scale 方式的训练会是不错的选择.



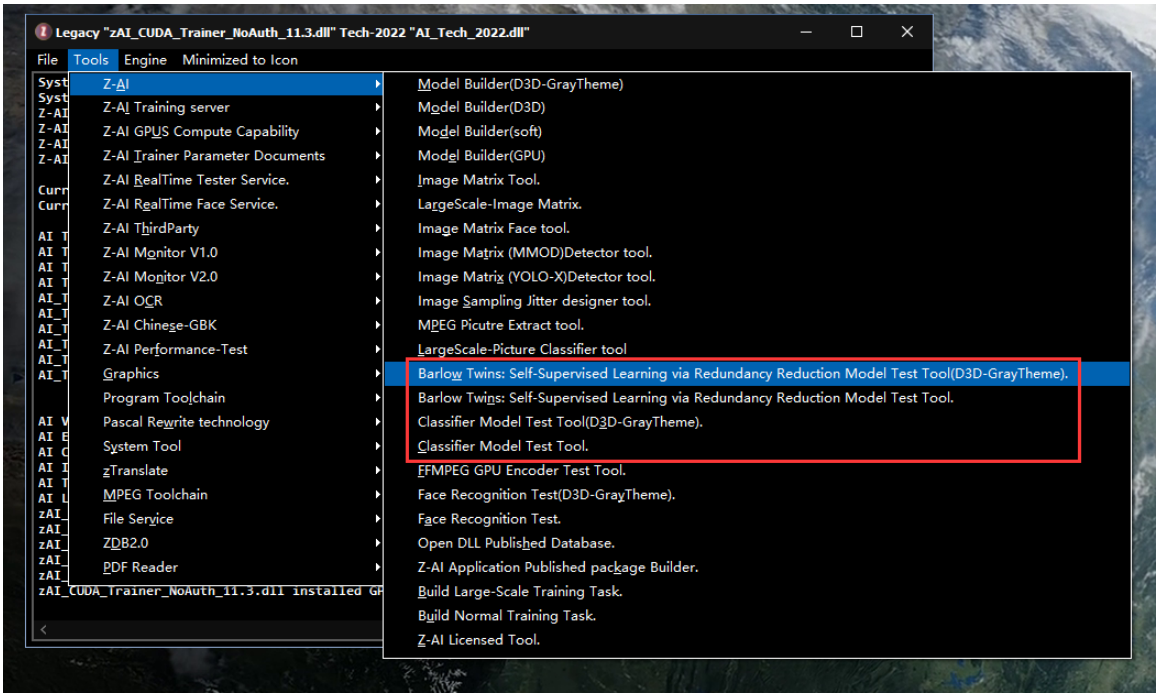
SSL 模型可以直接使用标注工具导图进来训练,无需任何标注,直接导图就行.



测试场景模型

,双击弹出 Z-AI 的工具链

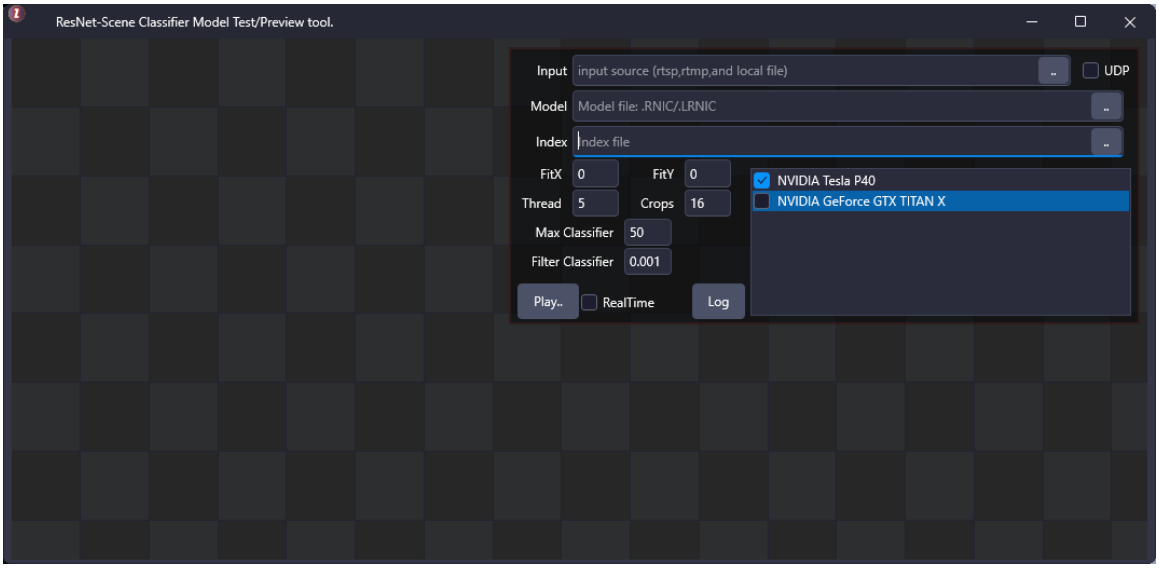
红框中就是场景模型测试,上面是 SSL,下面是 RNIC/LRNIC



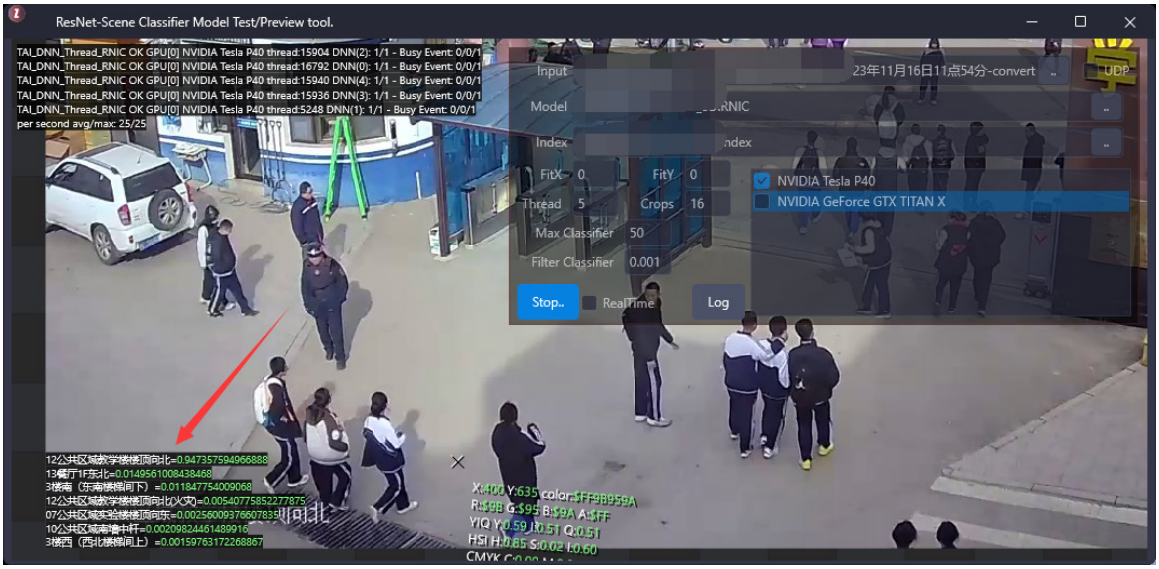
RNIC/LRNIC 测试工具

RNIC/LRNIC 的模型会随带一个.index 文本文件,这个文件会被建模工具生成

测试可以使用视频文件,推流地址,也可以是图片文件



在视频和图片的左下角输出识别结果

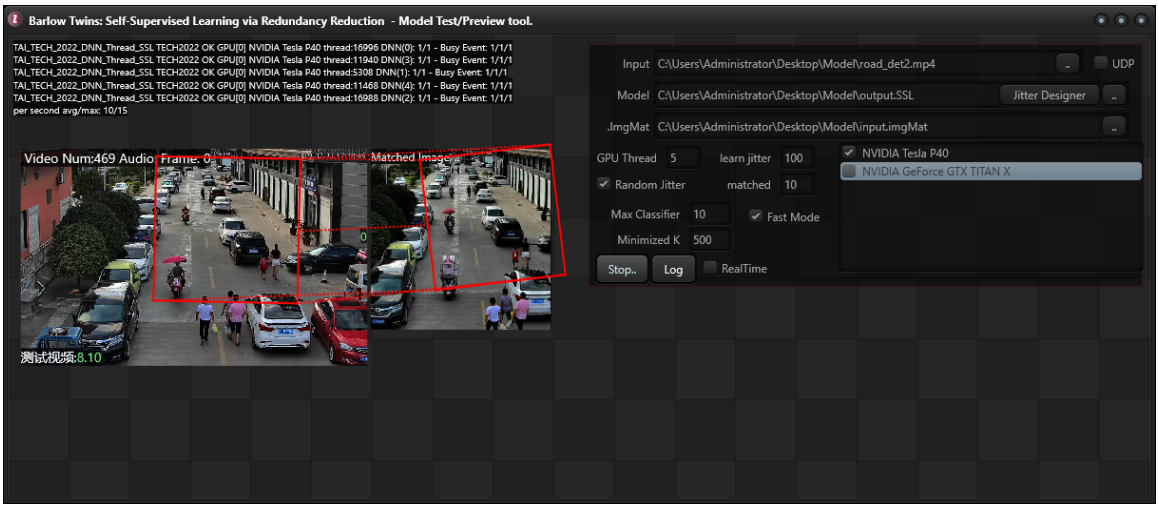


RNIC/LRNIC 的识别结果越大表示越接近场景目标分类,应用场景模型都是观测方式,在测试工具观测识别结果,然后在应用时限定识别结果,例如达到 0.8 就判断场景可以被识别,然后进入流程分支.

SSL 测试工具

SSL 面向场景识别的方式为,先指定 SSL 模型,然后指定一个场景样本库,然后 SSL 母体模型实时的对场景样本库建模,然后进入反推流程,目标视频会直接被反推到场景样本库中的某张图片上,通常来说,这是最接近的高速匹配

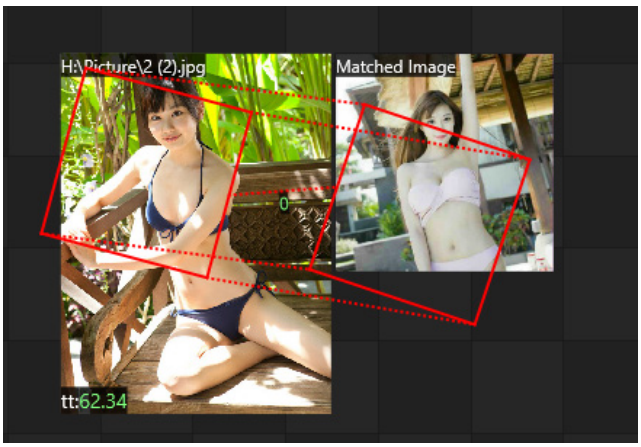
SSL 不光可以找出可以匹配的图片,还可以给出 jitter 匹配的区域,下图红框是机器观测出的相似区域.

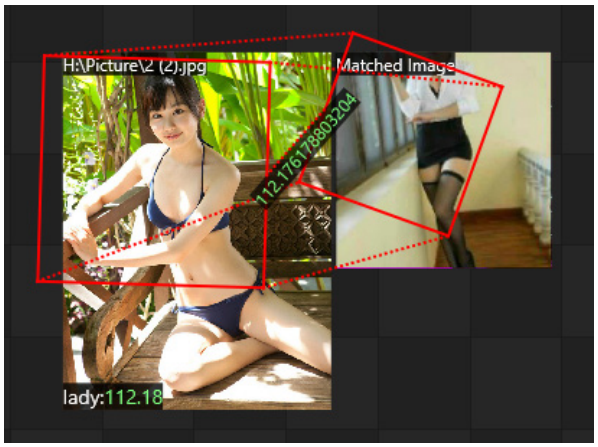


:SSL 的识别结果是相似差异值,该值越小就会越接近目标场景,这与 RNIC/LRNIC 是不同的.

在应用 SSL 模型时以 SSL 测试结果为准,通常 SSL 的差异值只要低于 100 就可以判定为高度匹配

测试图片在样本库不存在,这时候机器匹配的相似值大都会在 100 上下跳动





应用场景模型

场景模型高度依赖于框架,应该场景模型应该避免自己做 AI 框架,这会非常容易范错.

仔细阅读关于目标业务框架的文档,这些文档会对接入场景场景做详细详解,例如六代监控系统

全文完.

By qq600585

2024-3