1. 智慧交通的各类场景算法划分

1.1 智慧交通的各类场景划分

1.2 智慧交通的算法划分

1.3 智慧交通+AI安全功能

2. 智慧交通的落地部署架构方式

2.1 AI项目通用的开发交付流程

2.2 AI项目通用的部署形式

3. 边缘设备的行业场景应用

4. AI实战训练营课程分布

5. AI实战训练营奖励方案

6. 课程结束

# 1. 智慧交通的各类场景算法划分

## 1.1 智慧交通的各类场景划分

通常来说，智慧交通行业主要是对交通场景内的行人，机动车，非机动车进行识别分析。

行人识别分析包括对行人的姿态，方向，外观，以及基于行人的交通事件识别分析，（行人闯红灯等）。

机动车识别分析包括对机动车的外型，颜色，车灯，车窗，驾驶员安全事件分析（是否在打电话，是否系安全带等），车牌，车辆方向以及基于机动车的交通事件识别分析（超速检测，违停判定等）。

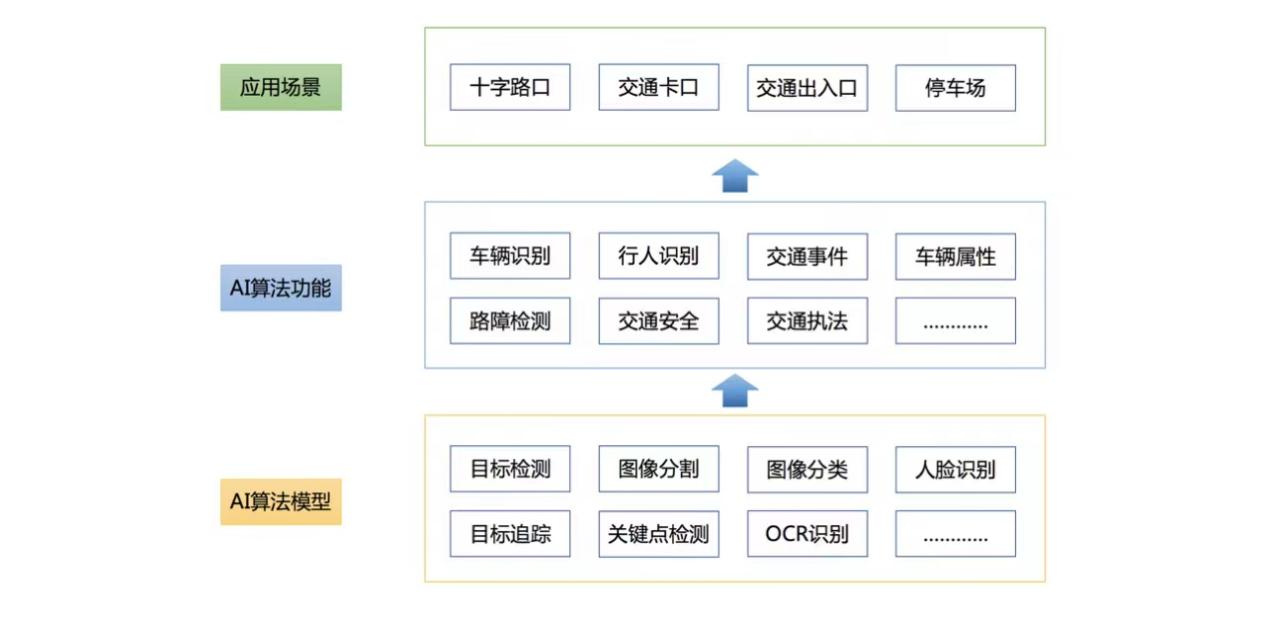
非机动车识别分析包括对非机动车的细分类别识别，运动状态识别，驾驶员安全事件分析（是否戴头盔等），以及基于非机动车的交通事件识别分析（非机动车闯红灯等）。



智慧交通的场景主要划分为：十字路口，交通卡口，交通出入口，停车场等场景。不同公司的业务领域、覆盖范围都不一样。有些会做交通场景的整体解决方案，有些会做算法提供商，还有一些会做交通细分领域的解决方案或者算法提供商。

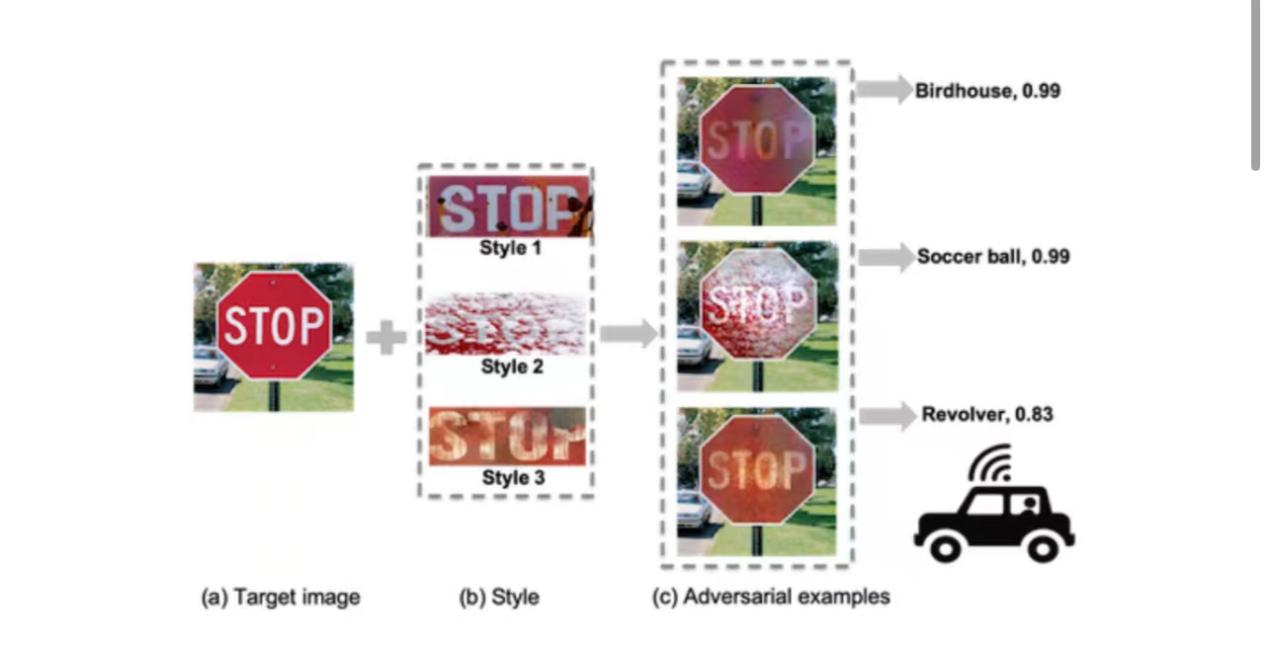
## 1.2 智慧交通的算法划分

在智慧交通的的场景中，每个场景都对应了不同的算法业务功能，每个算法业务功能则是由不同应用技术如目标检测，图像分割，图像分类，目标追踪，OCR识别等组合而成。

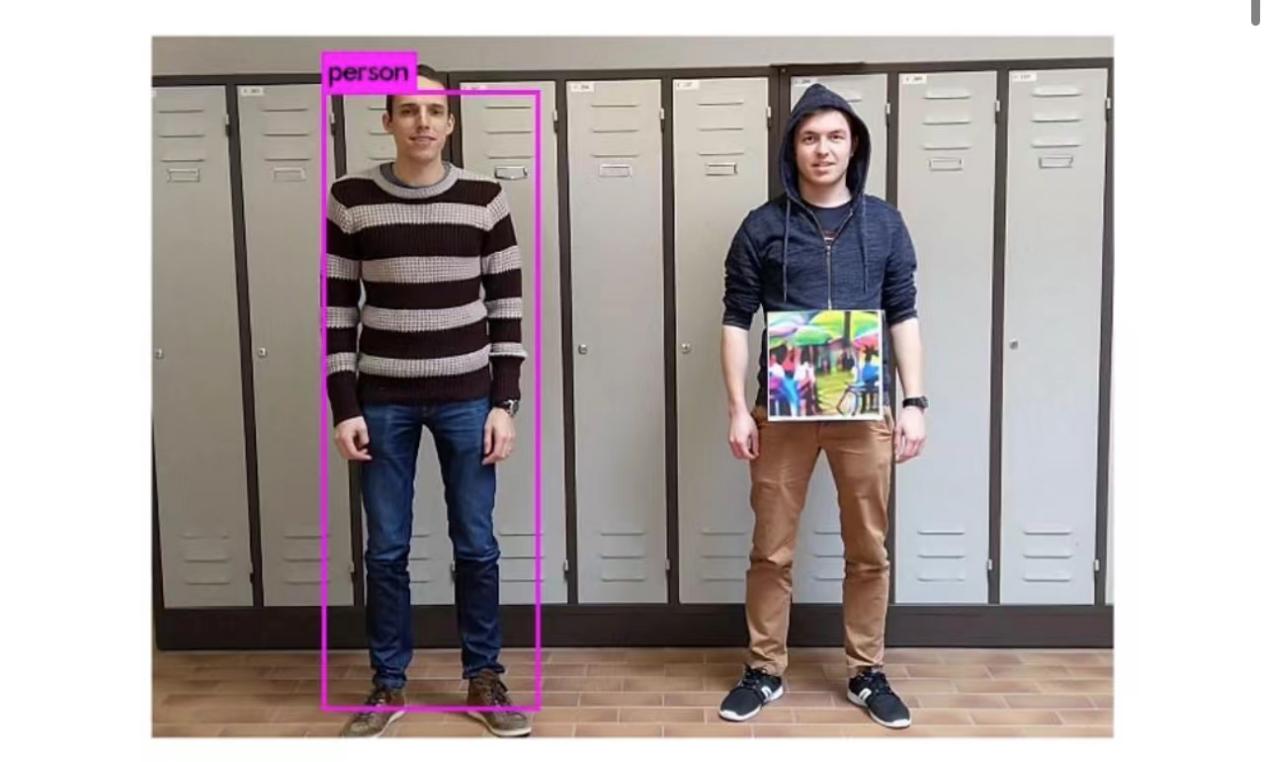


应用场景，AI算法功能，Ai算法模型这三者有机结合，形成了面对不同需求的整体算法解决方案。在上图里的AI算法功能中：车辆识别功能其底层的算法技术主要是车辆检测+车牌检测+车牌字符识别+业务功能判断等组成。这里的业务功能，包括车辆框矫正，车牌框矫正，车牌字符后处理等逻辑；行人识别算法功能，其底层的算法技术主要是行人检测+人脸识别+行人重识别等组成；车辆属性识别算法功能，其底层主要是车辆检测+车标识别+车系识别+驾驶员行为识别等组成。交通事件检测算法功能则更为复杂，其需要大量的交通规则作为约束，并且在检测机动车，行人以及非机动车的同时，要准确感知当前的背景环境条件。比如检测交通指示牌，检测交通道路指示线等来对交通事件的判定进行支持。行人闯红灯，车辆闯红灯，机动车超速，超时停车，机动车违停等都属于上述所说的交通事件中的典型细分功能。

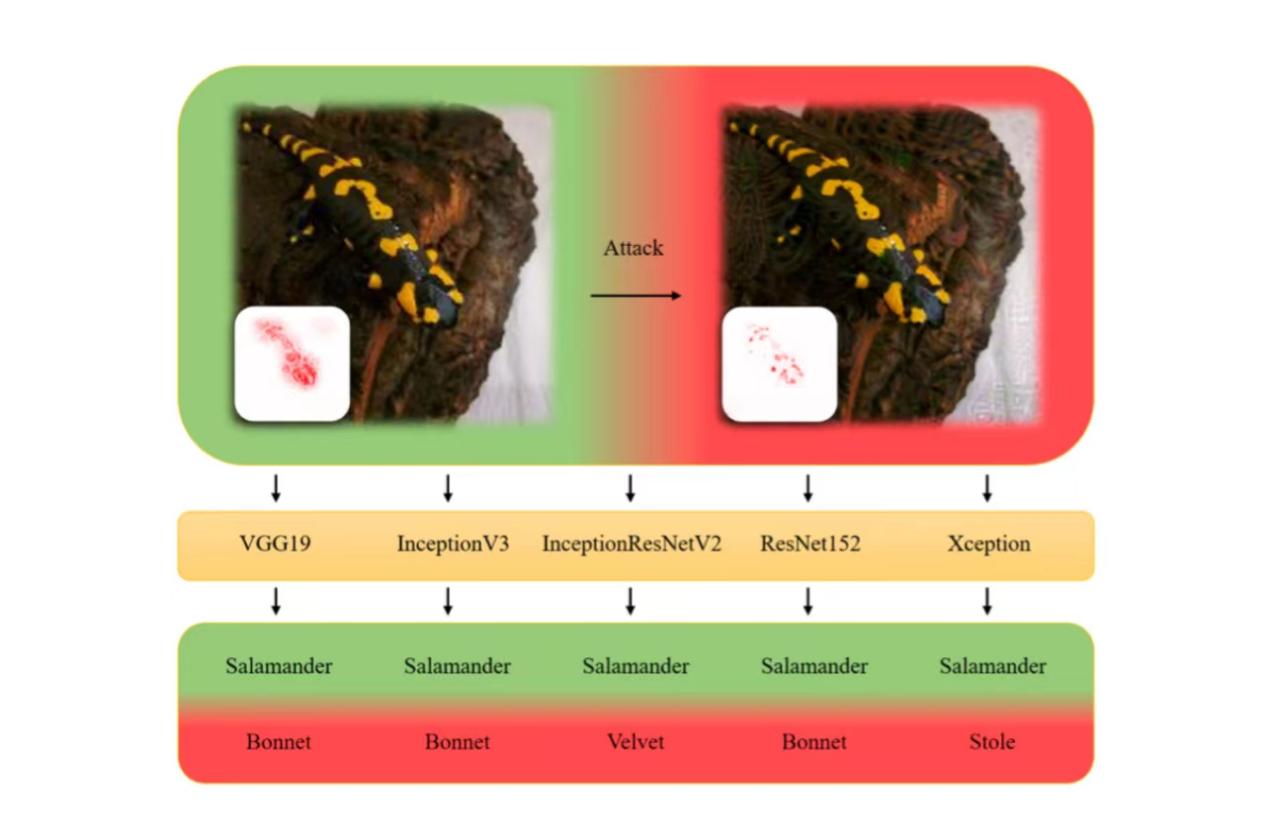
## 1.3 智慧交通+AI安全功能

经过上面的了解，大家可以发现，在智慧交通场景中，目标检测是AI项目里的前置任务。车辆识别，车辆属性识别，交通事件判定，交通执法判定，行人识别等智慧交通中的复合任务，都需要对感兴趣目标区域进行检测提取后才能接着进行后续的算法流程。然而，就在这些复合任务的流程中，可能会存在AI安全的风险。比如以本次训练营要讲的对抗攻击和对抗防御为例，对抗攻击能够让AI模型产生误判，从而可能引发严重的安全风险。

在对交通指示牌进行对抗攻击，主要有三种类似风格迁移的攻击模式，使得自动驾驶系统对STOP交通指示牌出现了误判，而每一种误判都可能引发严重的交通连锁后果。而智慧交通中的摄像头设备，也会接收此类对抗样本并将其作为后续算法功能的输入，从而导致算法解决方案中的某一环失效。



比如使用对抗贴纸让行人检测模型失效，在人体前放置一个对抗攻击算法特定生成的图画，便能对行人检测模型的性能产生较大的影响，无法对行人做出准确的检测。而当智慧交通中的摄像头设备，接收此类对抗样本并将其作为后续算法功能的输入，可能会导致包含行人检测的算法解决方案出现不确定风险。



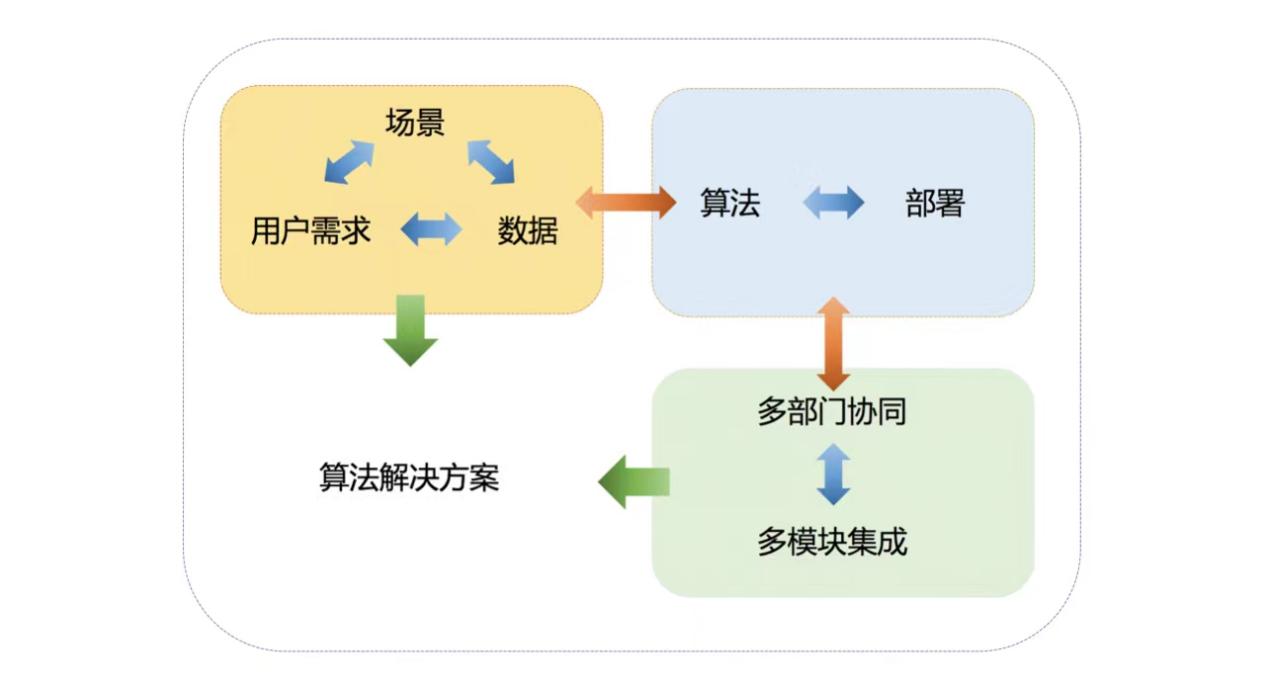
在图像分类中，可以看到对抗攻击算法攻击普通图片从而生成的对抗样本，会让目前主流的分类器出现明显的误分类，而图像分类算法是AI项目最基础的基石。面对上述的AI安全风险，我们需要设计使用一些AI安全防御策略，比如对抗样本监测，黑盒攻击防御，白盒攻击防御等，尽可能来防范可能存在的安全风险。后续的章节中也会从智慧交通中的分类场景入手，带着大家深入浅出的学习AI安全中关于对抗攻防的知识。

# 智慧交通的落地部署架构方式

讲完了智慧交通各个场景以及对应的算法解决方案和算法功能划分，分析了解了AI项目中可能存在的安全风险以及安全功能，接下来我们看看AI项目该如何开发交付。

## AI项目通用的开发交付流程

首先，我们来看一下AI项目通用的开发交付流程：



整个开发的流程需要分成以上的几个部分：

（1）明确场景&用户需求&获取数据

这三个维度是算法解决方案的起始，也是最关键的部分，正所谓“谋定而后动”。业务场景定义机器学习问题，数据侧不断去逼近这个机器学习问题的真实数据分布，而用户需求为我们指明了切入点和优化方向。比如车辆属性识别项目中，首先要确定场景，是十字路口，卡口，出入口还是停车场。在确定场景后，我们再与用户进行需求沟通，比如：哪些车辆属性是用户的核心需求，场景中的亮度，数据源（摄像头）获取的图像的清晰程度，是否存在大角度的目标，场景中的流量复杂程度等。 在完成了一系列的需求定义与评估之后，接着现场采集数据、数据标注、数据清洗校验等工作。

（2）算法训练&模型部署

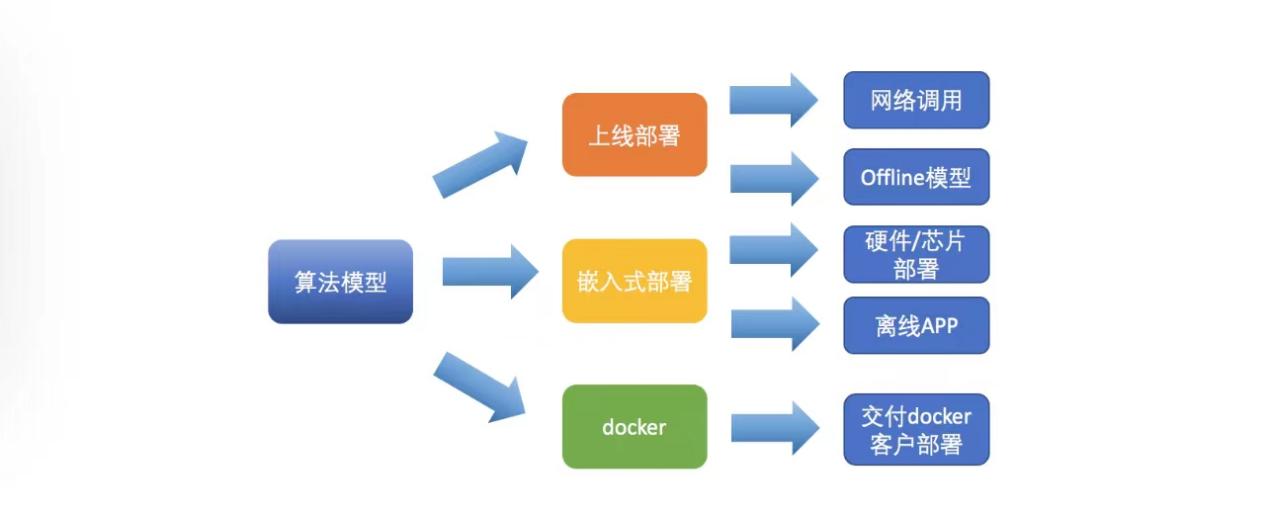
有了数据，我们再选用合适的算法模型，进行模型训练与优化。在车辆属性识别项目中，其底层主要是车辆检测 + AI安全模块 + 车标识别 + 车系识别 + 车牌识别 +驾驶员行为识别等一系列AI算法功能组成。完成模型的训练后，我们开始进行部署，主要包括模型转换、模型量化、模型与平台适配等工作。

（3）多部门协助&多模块集成

最后就是与不同部门进行协同，将多模块的算法功能与配套功能进行结合，形成完整的解决方案。

## AI项目通用的部署形式

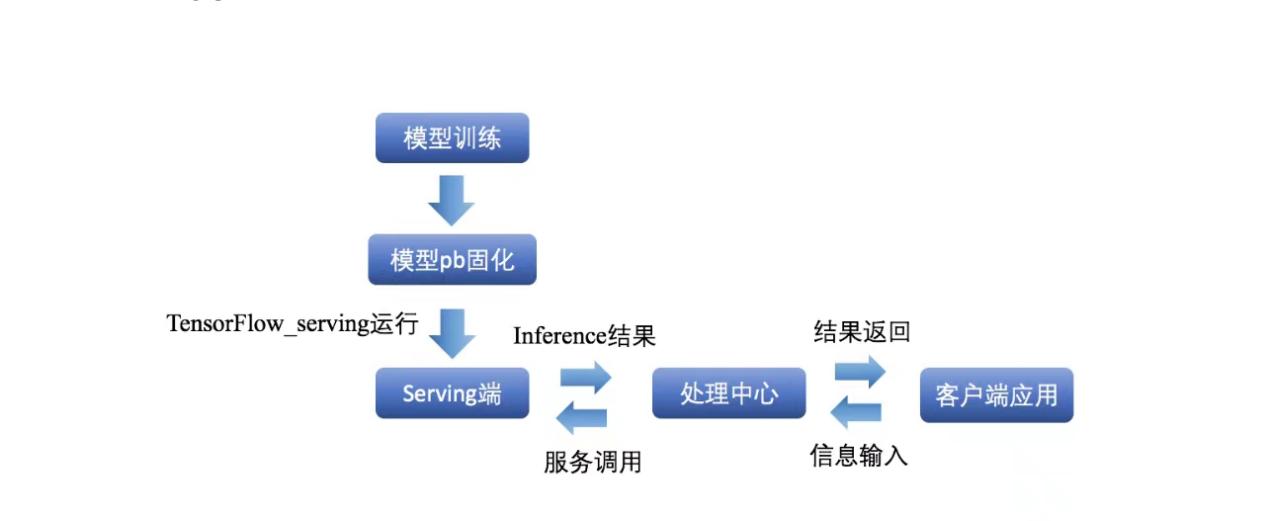
前面说到了模型部署，那么关于AI模型的部署形式，主要可以分为线上部署，嵌入式部署，交付docker三种：



（1）线上部署

在线上部署这个分支中，主要有网络调用与Offline模型两种情况。

如果是网络调用的形式，整个算法服务响应速度是非常重要的考虑因素。以TensorFlow框架的模型为例，一个有效的部署逻辑是：TensorFlow\_Serving + Docker + Web.

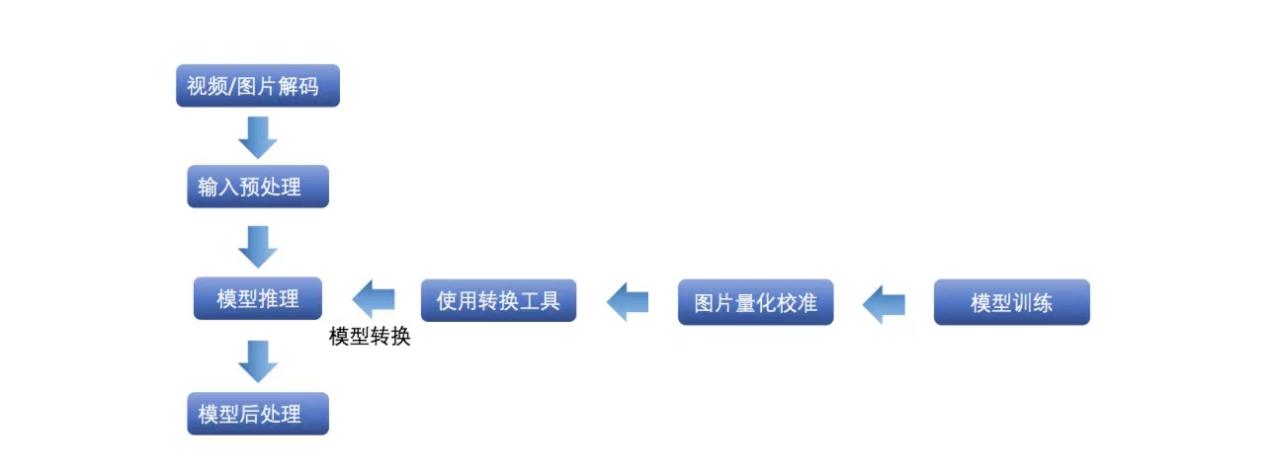


上图中web框架（flask、tornado、django）构成了这个客户端服务的处理中心，用来搭建http服务，将客户端输入传输给模型，并将模型输出结果打包返回。使用docker则避免了模型部署在不同服务器时产生不兼容的问题，因为docker相当于一个轻量级的虚拟环境，我们可以将模型所需的环境在docker中配置好，部署时只需启动docker即可。TensorFlow\_Serving能够很简单的把算法模型挂在服务器后台，当客户端侧有请求发送，它就会把模型运算后的结果打包返回。我们一般会用到两台服务器，其中一台作为TensorFlow\_Serving的服务器，专门用来部署模型并进行inference，通常为了效率，这个服务器一般是带GPU的；而另外一台服务器搭建处理中心，用来接收用户的输入以及其他业务的处理。在TensorFlow\_Serving的服务器上，TensorFlow\_Serving会自动根据传入的端口和模型路径进行部署操作。随后处理中心直接对模型所在服务器发起调用，调用代码可以使用java或python编写。综上，整个模型网络调用的链路就打通了。在实际项目中，比如车辆检测，我们将训练好的车辆检测模型，转换成pb格式，将整个运行环境在docker中配置好，并使用TensorFlow\_Serving工具将其挂在在模型专用服务器后台。而在客户服务器，我们使用web框架（flask、tornado、django）搭建客户端服务的处理中心。当客户需要获取现场的车辆检测结果时，通过发送请求给模型专用服务器，并通过TensorFlow\_Serving工具将模型输出的车辆检测结果打包返回给客户端。Offline模型：这种部署形式一般在不需要实时反馈的场景（精细化场景）中使用。在CV算法细分领域中，很多算法的赋能服务都是不需要实时的，这时可以使用Offline模型进行阶段性反馈。一个例子是交通场景的二次分析算法功能，当需要对一个交通事件或者违规事件进行复核时，需要使用相应的算法模型进行支持。而这个阶段更注重模型准确度与可靠性，对于实时性，耗时并没有太多要求。

（2）嵌入式部署

算法嵌入式部署也称为端侧部署，主要分为硬件/芯片部署与离线app两种形式。硬件/芯片部署中，一般都会进行模型的量化操作来优化模型。通常我们训练出来的模型是 FP32（32位浮点，单精度）的；而低精度主要有两种：FP16（半精度浮点）和INT8（8位的定点整数）。目前主流模型部署框架量化后的低精度一般是 INT8。这样可以在保证网络精度的基础上，大幅减少功耗，内存，传输延迟，大幅提高模型推理速度。使用Caffe框架训练模型并进行部署一般是硬件/芯片端使用模型最友好的方式，因为Caffe有原生C环境，与硬件平台的兼容性很好。而其他框架则需要使用特定的转换工具进行模型的转换从而能在特定平台上运行。

在实际场景中，我们在高性能端侧设备上能直接使用FP32的模型进行部署，而在低性能端侧设备上需要使用低精度INT8的模型进行部署。由于在进行精度转换时可能会存在转换后的精度变差的情况，所以需要在完成转换后，将低精度模型重新测试验证性能。总的来说，算法模型的嵌入式部署路径是：算法模型训练 -> 图片量化校准 -> 转换工具转换模型 -> 模型精度测试.



上图中在图片量化校准阶段，首先准备一个校准数据集，然后在校准数据集上进行FP32推理，每个网络层都寻找INT8精度下最优参数，最后从量化工具中输出INT8模型。离线app与硬件/芯片部署的不同主要体现在载体，离线app主要在手机端应用，即使用相应的手机端部署框架来进行模型转换。在芯片端，其一般都是基于ARM架构的，基于CPU完成深度神经网络推断的任务是最基础的方案，也是最可靠的。我们也可以用AI处理器进行网络推断，功耗和计算速度可以得到保证，速度也会比CPU快很多。而在带GPU的嵌入式端，也就是GPU盒子，使用CUDA可以加速的框架就能较好的部署，这也是最为方便的一种硬件部署方式。如果是手机端，还是需要使用一些专门的轻量级框架会比较适合。

（3）Docker部署

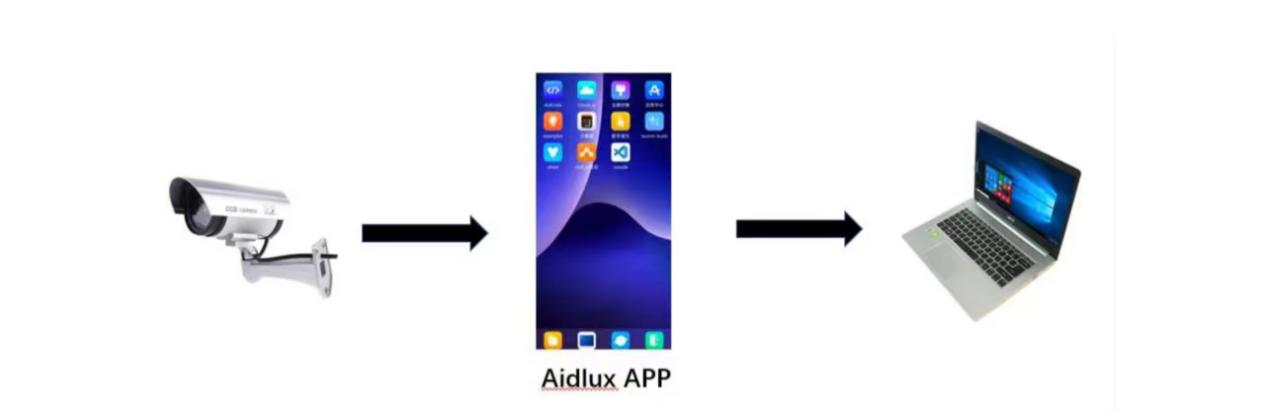
当一个公司不具备完整的算法上下游链路时，就单单只能作为算法模型API提供方，通过配置相应的docker镜像，交付给目标客户，让客户自己调取相应的算法接口，从而完成业务流程，这也就是交付docker，客户部署的情况。

# 边缘设备的行业场景应用

在智慧交通场景中，我们知道一般都是通过端侧设备的算力来支持AI算法的运行分析，并得到最后的处理结果。AI端侧设备主要由ARM架构的CPU+ GPU/TPU/NPU等协处理器 + 整体功耗 + 外围接口 + 工具链等部分组成，这也是算法工程师对端侧设备进行选型要考虑的维度。在实际业务中，根据公司的不同，算法工程师涉及到的硬件侧范围也会不一样。如果公司里硬件和算法由两个部门分别负责，那么算法工程师最多接触到的硬件侧知识就是硬件性能评估，模型转换与模型硬件侧验证，一些硬件高层API接口的开发与使用；如果公司里没有这么细分的部门，那么算法工程师可能就会接触到端侧的视频编解码，模型推理加速，Opencv，FFmpeg，Tensor RT，工具链开发等角度的知识。



那么算法工程师该如何看待硬件侧呢？首先，整体上还是要将硬件侧工具化，把端侧设备当做算法模型的一个下游载体，会熟练的选型与性能评估更加重要。端侧设备是算法产品整体解决方案中一个非常重要的模块，算法+硬件的范式将在未来的边缘计算与万物智能场景中持续发力。在日常业务中，算法模型与端侧设备的适配性与兼容性是必须要考虑的问题，端侧设备是否兼容一些特殊的网络结构？算法模型转化并部署后，精度是否下降？功耗与耗时能否达标？等等都是我们需要考虑的维度。在本训练营中，为了方便大家尝试学习，我们使用了平板/手机 + Aidlux边缘设备软件 + 电脑的形式。Aidlux软件可以在各大安卓应用商店中找到并下载，为大家提供了一个可以在安卓手机/平板上开发和部署的环境，并且内置了各种AI案例参考，这个环境和现实中的边缘计算设备基本一致。学会熟悉了之后，以后在真实项目中使用边缘设备就能够事半功倍。



不过需要注意的是，这里的Aidlux因为是安装在ARM架构下的，跨生态（Android/鸿蒙+Linux）一站式开发和部署平台APP。必须要在安卓手机或者安卓的平板上才能下载运行，大家在第三节课之前最好准备一种安卓设备。

# 4 AI实战训练营课程分布

为了带大家整体学习与实现各个功能，本次训练营的课程主要分成6节课。

第一节课，即本节课，主要讲解智慧交通行业的目前状况，并从AI项目部署架构，边缘设备、AI安全应用等内容进行讲解。

第二节课，在11.25晚上8点，主要从AI安全风险的角度展开，对AI安全技术以及“以数据为中心”策略进行讲解，并对Aidlux的各个方面内容进行尝试。

第三节课，在11.28号晚上8点，主要从目标检测算法的训练，Aidlux上的移植、测试等方面进行讲解。

第四节课，在11.30号晚上8点，主要对AI安全中的对抗攻击与对抗防御进行讲解，并在PC端和Aidlux端进行部署与测试。

第五节课，在12.02号晚上8点，主要对AI安全项目功能进行讲解，以车辆检测+ AI对抗攻防+ 后置分类任务为主线，完成AI业务与AI安全相结合的过程。

第六节课，也就是最后一节课，在12.06号晚上8点，会举办一次结营活动，对于完成训练营作业的同学办法获奖证书，以及一些列的礼物奖励。

# 5 AI实战训练营奖励方案

智慧交通完成训练的同学，也会有很多的结营奖励。

一是结业证书，完成作业的同学都可以获得训练营晋级的荣誉结业证书。

二是有机会晋级Aidlux认证开发者，拥有更多的福利，比如价值数千元的AidBox边缘设备一台。

此外，讲师和内容评审团，会针对大家作业的内容完成度得分+课堂表现分+加分项，评选出20名给予红包奖励，一共3000元。

第一名500元

第2名400元

第3名300元

第4名200元

第5~20名 100元。

在此过程中，如果有任何的问题，都可以添加Aidlux小助手的微信。本节课的内容到这里就结束了，第一节涉及代码层面的不多，主要是先从智慧交通整体架构&CV项目落地应用方面和大家一起了解行业现状，并在中间穿插了部分AI安全的相关知识，大家也可以慢慢温习下本节课的内容。感谢大家本节课的学习，有相关的问题，也可以在群中一起交流沟通。