

华为 HiLens

用户指南

文档版本 01
发布日期 2025-02-24



版权所有 © 华为技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 用户介绍	1
2 使用流程简介	2
3 权限管理	7
3.1 创建用户并授权使用华为 HiLens	7
3.2 华为 HiLens 自定义策略	8
3.3 权限策略和授权项	9
3.3.1 权限及授权项说明	9
3.3.2 设备管理权限	10
3.3.3 技能开发权限	11
3.3.4 技能市场权限	13
3.3.5 管理产品权限	13
4 创建工作空间	15
5 注册 HiLens Kit	20
5.1 HiLens Kit 注册流程	20
5.2 使用 SSH 注册设备	22
5.2.1 连接 PC 和 HiLens Kit	23
5.2.2 升级 HiLens Kit 系统固件版本	25
5.2.3 SSH 登录 HiLens Kit 设备	27
5.2.4 无线网络配置（SSH）	29
5.2.5 有线网络配置（SSH）	33
5.2.6 SSH 注册 HiLens Kit	36
5.3 组网配置	38
5.3.1 组网配置方式	38
5.3.2 无线网络（外网）&有线网络（内网）	39
5.4 配置防火墙	44
5.5 使用 SD 卡	45
6 管理设备	46
6.1 设备管理简介	46
6.2 管理设备上的技能	46
6.2.1 技能管理简介	46
6.2.2 安装技能	48

6.2.3 添加运行时配置.....	49
6.2.4 配置订阅消息.....	50
6.2.5 设置难例.....	53
6.2.6 启动或停止技能.....	54
6.2.7 卸载技能.....	55
6.3 查看设备信息.....	56
6.4 查看设备日志.....	56
6.5 升级 HiLens_Device_Agent 固件版本.....	57
6.6 配置摄像头.....	58
6.7 注销设备.....	60
6.8 查看设备告警.....	61
6.9 一键健康检查.....	63
7 控制台开发技能.....	65
7.1 技能简介.....	65
7.2 新建技能.....	68
7.2.1 使用技能模板.....	68
7.2.2 使用空模板.....	70
7.3 获取技能模板.....	77
7.4 管理算法模型.....	79
7.4.1 开发算法模型.....	79
7.4.2 导入（转换）模型.....	80
7.4.3 模型输入目录规范.....	85
7.4.4 模型输出目录规范.....	86
7.5 编写逻辑代码.....	86
7.6 编辑技能.....	87
7.7 安装和调试技能.....	87
7.8 删除技能.....	88
8 技能市场.....	90
8.1 技能市场简介.....	90
8.2 在技能市场查找技能.....	92
8.3 购买技能.....	92
8.4 管理订单.....	93
9 管理数据.....	95
10 管理产品.....	99
10.1 产品简介.....	99
10.2 新建产品.....	99
10.3 订购技能（用于 35XX 系列设备）.....	102
10.4 分发技能.....	102
10.5 新增或删除技能.....	104
A Caffe 算子边界.....	106

B Tensorflow 算子边界	135
--------------------------------	------------

1 用户介绍

华为HiLens是面向普通用户、AI应用开发者、硬件厂商的端云协同AI应用开发平台。它由具备AI推理能力的摄像头和云上开发平台组成，包括一站式技能开发、部署与设备管理、技能市场等功能，帮助用户开发AI技能并将其推送至端侧设备。

针对不同类型的用户的不同使用场景，其关注和使用的华为HiLens的功能有所区别。首先，请通过[表1-1](#)介绍了解不同角色及其使用场景，三种不同场景的使用流程请参见[使用流程简介](#)。

表 1-1 用户介绍

用户角色	典型用户	用户场景	使用流程
普通用户	<ul style="list-style-type: none">家庭成员商超老板工地负责人	<ul style="list-style-type: none">家庭：提高家庭安防。商超：统计客流量等。园区：检测车辆属性和车牌等。工地：检测工人是否佩戴安全帽等。	参见 普通用户使用技能
开发者	<ul style="list-style-type: none">从事AI开发的技术人员高校学生	开发具备AI能力的技能，发布到技能市场，构建良好AI开发者生态。 <ul style="list-style-type: none">控制台新建技能	参见： 开发者控制台开发技能
厂商	海思35xx系列芯片摄像头产品的厂商	<ul style="list-style-type: none">摄像头厂商：赋予中低端摄像头设备AI能力。	参见 厂商管理产品

2 使用流程简介

华为HiLens面向三种不同用户提供了设备管理、技能开发、数据管理、技能市场、产品管理等功能，帮助用户开发AI技能并将其推送到端侧设备。

说明

端侧设备HiLens Kit的具体使用方法请参见[HiLens Kit用户指南](#)。

基于不同的使用场景，用户所需的功能不同。针对三种不同角色及其使用场景，梳理了其在华为HiLens中的主要使用流程；另外，针对华为HiLens提供的功能，您可以通过功能导读快速了解功能并通过链接跳转至相应章节了解详情。

针对不同角色的使用流程如下所示，此处仅罗列主要流程，其他管理操作可参见[华为HiLens功能导读](#)。

- [普通用户使用技能](#)
- [开发者控制台开发技能](#)
- [厂商管理产品](#)

前提条件

- 在使用华为HiLens之前您需要申请华为云账号。通过此账号，您可以使用所有华为云服务，并且只需为您所使用的服务付费。
- 由于华为HiLens服务依赖其他服务，所以需要在开始使用华为HiLens前获得相关服务的权限，包含ModelArts、OBS和SWR服务，华为HiLens与相关服务的关系请参见[与其他服务的关系](#)。

普通用户使用技能

普通用户是指使用HiLens Kit设备，希望提高家庭、商超、园区或工地等场景下的智能监控能力的用户。

图 2-1 普通用户使用流程

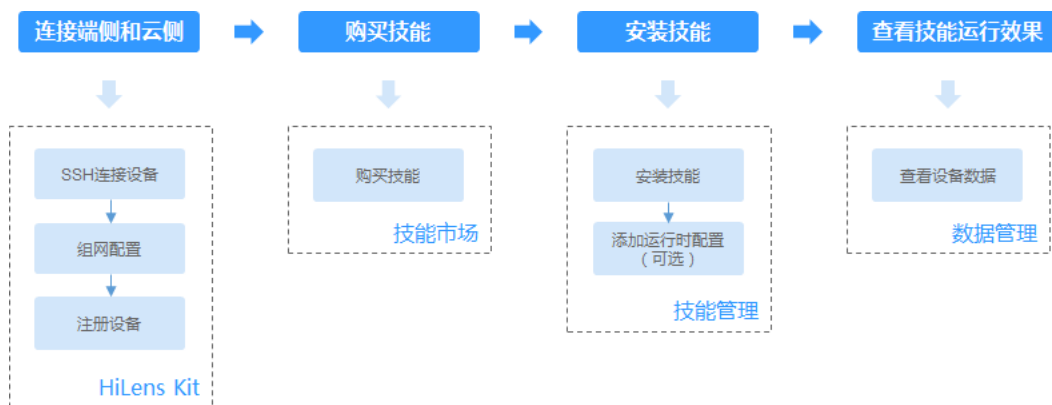


表 2-1 普通用户使用流程说明

主任务	说明	详细指导
注册HiLens Kit	首先，连接您购买的HiLens Kit，并将HiLens Kit注册到华为HiLens平台，连接端侧与云侧。	HiLens Kit注册流程 使用SSH注册设备
升级HiLens_Device_Agent固件版本	升级HiLens_Device_Agent固件版本至1.0.7及其以上版本。	升级HiLens_Device_Agent固件版本
订购技能	在技能市场直接购买所需要的技能，或者在技能市场上购买已定制完成的技能，以便安装在设备上使用，拓展设备的AI能力。在技能市场中，请选择适用于Ascend芯片的技能。	购买技能
安装技能	将购买的技能安装到设备上。	安装技能
查看技能运行效果	查看设备数据以及技能的运行效果。	管理数据

开发者控制台开发技能

华为HiLens提供了技能开发的平台，您可以根据自身情况开发可用于Ascend310或者海思35XX系列的技能。适用于Ascend310芯片的技能可直接安装部署在HiLens Kit设备中使用。

由于使用海思35XX系列芯片的设备，其芯片内存和性能偏低，需要对模型进行优化后才能运行，其对应的技能开发难度较大，如果碰到困难，可联系华为HiLens平台工作人员支撑。

- 技能开发过程中需要将技能部署到设备，以便查看技能运行效果，所以建议开发者购买一个HiLens Kit设备。
- 开发技能需要模型，所以开发者需要在本地或在ModelArts中训练好AI模型。在ModelArts中训练模型可参见ModelArts训练模型。

使用流程如图2-2，流程说明请参见表2-2。

图 2-2 开发者使用流程

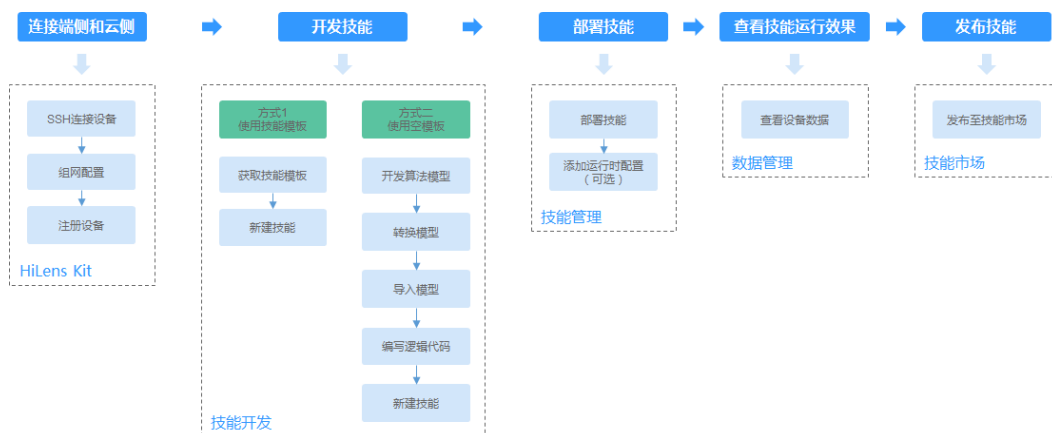


表 2-2 开发者使用流程说明

流程	说明	详细指导
注册设备	首先，将设备注册到华为HiLens平台，连接端侧与云侧。华为HiLens支持HiLens Kit、Atlas500、Atlas 800，IVS 1800以及其他适配中设备。	注册HiLens Kit
开发技能	使用技能模板新建技能： 获取华为HiLens提供的技能模板（已经包含了算法模型和逻辑代码），快速新建技能。	获取技能模板 使用技能模板
	使用空模板新建技能： 为满足更多业务诉求，可以自行开发模型，并将开发好的算法模型导入华为HiLens，根据业务诉求编写逻辑代码，然后基于您自定义的算法模型和逻辑代码新建技能。 如果您的算法模型不符合华为HiLens格式要求，还可以使用模型转换功能满足要求。	开发算法模型 导入（转换）模型 编写逻辑代码 使用空模板
部署技能	将技能部署到设备上，便于调试运行效果。	安装和调试技能
查看技能运行效果	查看设备数据以及技能的运行效果。	管理数据

厂商管理产品

针对生产海思35xx系列芯片的摄像头的设备商，华为HiLens提供了一个产品管理的流程。设备厂商可以通过华为HiLens管理设备产品，订购技能，然后将License分发至产品设备，使得厂商的设备具备AI能力。

厂商管理产品的使用流程如图2-3，流程说明请参见表2-3。

图 2-3 厂商使用流程

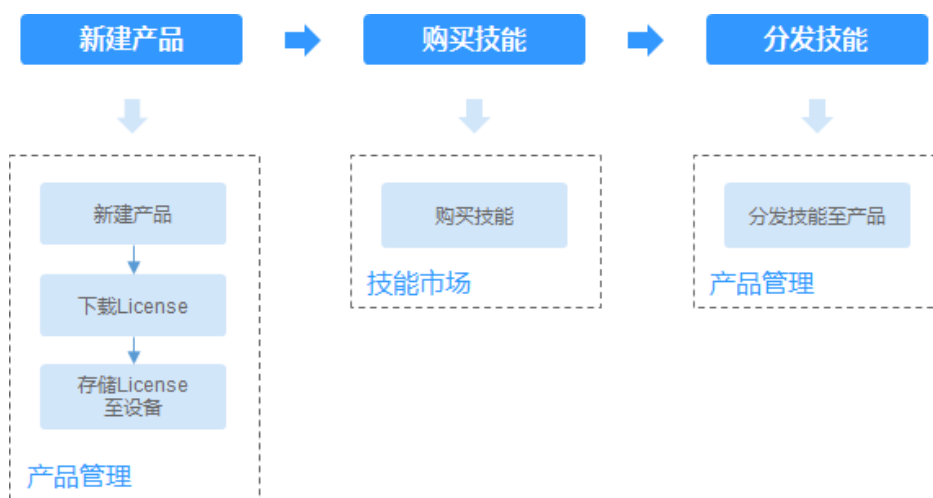


表 2-3 厂商的使用流程说明

流程	说明	详细指导
新建产品	厂商创建一个产品，然后把设备“关联”到这个产品。	新建产品
订购技能	在技能市场直接购买所需要的技能（适用于海思35XX系列的芯片），以便分发至用户设备上使用。	购买技能
分发技能	将订购的技能分发到设备上，并下载 SDK集成到设备上，供用户使用。	分发技能

华为 HiLens 功能导读

表 2-4 文档导读

章节	面向角色	说明
注册帐号	ALL	首先，作为任何一种角色，您需要申请华为云账号。通过此账号，您可以使用所有华为云服务，并且只需为您所使用的服务付费。在开始使用华为HiLens前获得其他相关服务的授权，然后再进入华为HiLens管理控制台使用相关功能。
注册HiLens Kit	<ul style="list-style-type: none"> 普通用户 开发者 	针对普通使用者或开发者，在使用技能或开发技能前，需先购买端侧设备，并且参考操作指导完成连接，将端侧设备与云侧华为HiLens管理控制台完成连接。

章节	面向角色	说明
设备管理	<ul style="list-style-type: none"> 普通用户 开发者 	您可以通过华为HiLens管理控制台管理已注册的端侧设备。包括设备上的技能管理、升级固件、配置摄像头等操作。对于拥有端侧设备的用户（包括普通使用者和开发者），您都可以通过设备管理的操作，管理您的设备。
技能开发	开发者	<p>对于开发者，您可以在华为HiLens平台新建您的技能，可以使用技能模板新建技能，或者自行开发算法模型和逻辑代码，再用于新建技能。另外，还支持将您开发的技能安装部署至设备进行调试。</p> <p>最后，针对调试完成的技能，您可以将其发布至技能市场，供更多用户获取使用，同时您还能获得相应的报酬。</p>
技能市场	ALL	技能市场是一个开放的平台，提供了适用于不同芯片、不同应用场景的技能。您可以在技能市场购买技能。如果您是开发者，也可以将自己开发的技能发布至技能市场，收取相应的报酬。
数据管理	<ul style="list-style-type: none"> 普通用户 开发者 	针对注册至华为HiLens的设备，您可以查看设备的视频数据以及技能的运行效果。通过查看设备数据，了解技能运行效果。
管理产品	厂商	针对厂商（使用海思35XX系列芯片的设备厂商），您可以使用华为HiLens平台管理您的产品，同时订购相应的技能，将技能分发至产品中，并下载SDK集成到设备上，使得您的设备产品具备AI能力。

3 权限管理

3.1 创建用户并授权使用华为 HiLens

如果您需要对您所拥有的华为HiLens进行精细的权限管理，您可以使用[统一身份认证服务](#)（Identity and Access Management，简称IAM），通过IAM，您可以：

- 根据企业的业务组织，在您的华为云账号中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用华为HiLens资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将华为HiLens资源委托给更专业、高效的其他华为云账号或者云服务，这些账号或者云服务可以根据权限进行代运维。

如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用华为HiLens服务的其它功能。

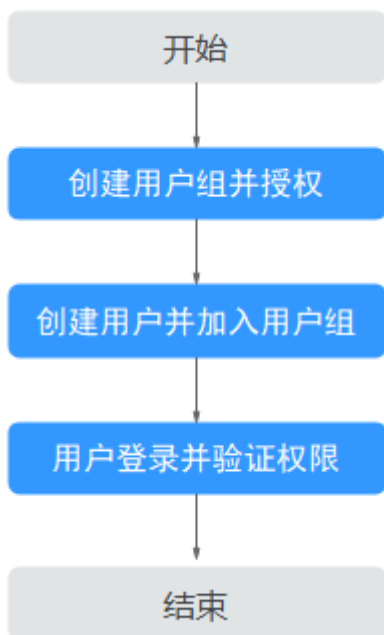
本章节为您介绍对用户授权的方法，操作流程如[图3-1](#)所示。

前提条件

给用户组授权之前，请您了解用户组可以添加的华为HiLens权限，并结合实际需求进行选择，华为HiLens支持的系统权限，请参见：[HiLens权限管理](#)。如果您需要对除华为HiLens之外的其它服务授权，IAM支持服务的所有权限请参见[权限策略](#)。

示例流程

图 3-1 给用户授予华为 HiLens 权限流程



1. 创建用户组并授权

在IAM控制台创建用户组，并授予HiLens CommonOperations权限。如果需要申请公测、设置告警接收和设置技能消息的操作权限，需要授予SMN Administrator系统角色。

2. 创建用户并加入用户组

在IAM控制台创建用户，并将其加入1中创建的用户组。

3. 用户登录并验证权限

新创建的用户登录控制台，切换至授权区域，验证权限：

在“服务列表”中选择华为HiLens，进入华为HiLens主界面，单击“设备管理>设备列表”，选择其中一个设备卡片，单击“操作>注销”（如果注册设备后需要激活设备，请先退订管理费后再执行注销操作），如果无法注销设备（假设当前权限仅包含HiLens CommonOperations），表示“HiLens CommonOperations”已生效。

3.2 华为 HiLens 自定义策略

如果系统预置的华为HiLens权限，不满足您的授权要求，可以创建自定义策略。自定义策略中可以添加的授权项（Action）请参考[权限及授权项说明](#)。

目前华为云支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。
- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

具体创建步骤请参见：[创建自定义策略](#)。本章为您介绍常用的华为HiLens自定义策略样例。

华为 HiLens 自定义策略样例

示例：拒绝用户删除已开发的技能

拒绝策略需要同时配合其他策略使用，否则没有实际作用。用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循Deny优先。

如果您给用户授予华为HiLens FullAccess的系统策略，但不希望用户拥有华为HiLens FullAccess中定义的删除已开发的技能权限，您可以创建一条拒绝删除已开发的技能的自定义策略，然后同时将华为HiLens FullAccess和拒绝策略授予用户，根据Deny优先原则，则用户可以对华为HiLens执行除了删除已开发的技能外的所有操作。拒绝策略示例如下：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "hilens:project:deleteProject"
      ]
    }
  ]
}
```

3.3 权限策略和授权项

3.3.1 权限及授权项说明

如果您需要对您所拥有的华为HiLens服务进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用华为HiLens服务的其它功能。

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略或角色，才能使用户组中的用户获得相应的权限，这一过程称为授权。授权后，用户就可以基于已有权限对云服务进行操作。

权限根据授权的精细程度，分为**角色**和**策略**。角色以服务为粒度，是IAM最初提供了一种根据用户的工作职能定义权限的粗粒度授权机制。策略以API接口为粒度进行权限拆分，授权更加精细，可以精确到某个操作、资源和条件，能够满足企业对权限最小化的安全管控要求。

说明

如果您要允许或是禁止某个接口的操作权限，请使用策略。

账号具备所有接口的调用权限，如果使用账号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来查询云服务器列表，那么这个IAM用户被授予的策略中必须包含允许“hilens:servers:list”的授权项，该接口才能调用成功。

支持的授权项

策略包含系统策略和自定义策略，如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝某项操作。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。
- 授权范围：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。关于IAM项目与企业项目的区别，详情请参见：[IAM与企业管理的区别](#)。
- 对应API接口：自定义策略实际调用的API接口。

华为HiLens服务的支持自定义策略授权项如下所示：

- [设备管理权限](#)
- [技能开发权限](#)
- [技能市场权限](#)
- [管理产品权限](#)

3.3.2 设备管理权限

表 3-1 设备管理

权限说明	方法	授权项	依赖其他服务的最小化授权项
注销设备	PUT	hilens:device:deleteDevice	-
升级设备固件	PUT	hilens:device:upgradeDeviceFirmware	-
注册设备	POST	hilens:device:registerDevice	-
更新设备信息	PUT	hilens:device:updateDevice	-
更新设备配置信息	PUT	hilens:device:updateDeviceConfig	-
获取设备配置信息	GET	hilens:device:getDeviceConfig	-
获取设备列表	GET	hilens:device:getDeviceList	-

权限说明	方法	授权项	依赖其他服务的最小化授权项
获取设备下已安装的技能列表	GET	hilens:device:getDeviceSkillList	-
卸载设备上指定技能	DELETE	hilens:skill:uninstallSkill	-
更新设备上指定技能的技能配置信息	PUT	hilens:skill:updateSkillConfig	-
启动设备上指定技能	POST	hilens:skill:startSkill	-
停止设备上指定技能	POST	hilens:skill:stopSkill	-
获取设备上指定技能的技能配置信息	GET	hilens:skill:getConfig	-

3.3.3 技能开发权限

表 3-2 技能开发管理

权限说明	方法	授权项	依赖其他服务的最小化授权项
创建模型转换作业	PUT	hilens:model:convertModel	-
发布已开发的技能到技能市场	PUT	hilens:project:publishProject	obs:object:GetObject obs:object:PutObject obs:object:PutObjectACL
下载技能模板	PUT	hilens:template:downloadTemplate	obs:object:GetObject
删除技能模型	DELETE	hilens:model:deleteModel	-
创建技能	POST	hilens:project:createProject	obs:object:GetObject obs:bucket:HeadBucket obs:bucket:CreateBucket obs:object:PutObject obs:bucket>ListAllMyBuckets obs:bucket>ListBucket

权限说明	方法	授权项	依赖其他服务的最小化授权项
编辑已开发的技能	PUT	hilens:project:updateProject	obs:object:GetObject obs:bucket:HeadBucket obs:bucket:CreateBucket obs:object:PutObject obs:bucket:ListAllMyBuckets obs:bucket:ListBucket
导入技能模型到华为HiLens服务	POST	hilens:model:importModel	obs:object:GetObject obs:bucket:HeadBucket obs:bucket:ListAllMyBuckets obs:bucket:ListBucket
删除已开发的技能	DELETE	hilens:project:deleteProject	-
收藏技能模板	POST	hilens:template:collectTemplate	-
部署已开发的技能到设备上	PUT	hilens:project:deployProject	obs:object:GetObject
取消已收藏的技能模板	DELETE	hilens:template:deleteCollectedTemplate	-
获取模型转化作业详情	GET	hilens:model:getConvertJob	-
获取技能模板详情	GET	hilens:template:getTemplate	-
获取技能模型详情	GET	hilens:model:getModel	-
获取已开发的技能详情	GET	hilens:project:getProject	-
获取模型转化作业列表	GET	hilens:model:getConvertJobList	-
获取已开发的技能列表	GET	hilens:project:getProjectList	-
获取技能模型列表	GET	hilens:model:getModelList	-
获取技能模板列表	GET	hilens:template:getTemplateList	-

3.3.4 技能市场权限

表 3-3 技能市场管理

权限说明	方法	授权项	依赖其他服务的最小化授权项
下载技能市场的技能	GET	hilens:market:downloadSkill	obs:object:GetObject
创建技能市场的技能订单	POST	hilens:market:createSkillOrder	-
安装技能市场购买的技能	POST	hilens:market:installSkill	-
下架已发布在技能市场的技能	PUT	hilens:market:withdrawSkill	-
获取技能计费信息	GET	hilens:market:getSkillBillingInfo	-
获取技能市场的技能详情	GET	hilens:market:getSkillInfo	-
获取技能市场的技能订单列表	GET	hilens:market:getSkillOrderList	-
获取技能市场列表	GET	hilens:market:getSkillList	-

3.3.5 管理产品权限

表 3-4 管理产品

权限说明	方法	授权项	依赖其他服务的最小化授权项
分发技能订单至产品	POST	hilens:product:createProductLicense	-
更新产品下技能订单信息	PUT	hilens:product:updateProductLicense	-
删除产品	DELETE	hilens:product:deleteProduct	-
删除产品下技能订单信息	DELETE	hilens:product:deleteProductLicense	-

权限说明	方法	授权项	依赖其他服务的最小化授权项
创建产品	POST	hilens:product:createProduct	-
更新产品信息	PUT	hilens:product:updateProduct	-
获取产品列表	GET	hilens:product:getProductList	-
获取产品下技能订单列表	GET	hilens:product:getProductLicenseList	-

4 创建工作空间

在使用华为HiLens之前，如果您想实现资源隔离管理，您可以创建工作空间，实现资源隔离，即在不同工作空间导入的模型、创建的技能等资源，均实现隔离管理。

隔离资源说明

每个华为云账号或IAM账号，均存在一个默认的工作空间“default”，该空间可查看当前账号下在华为HiLens所创建的所有资源。

创建工作空间后，可实现隔离管理的资源如下：

表 4-1 隔离资源

导航栏	隔离资源
设备管理>设备列表	注册的HiLens Kit以及设备上的技能管理、摄像头管理、告警管理、固件版本资源。
技能开发>模型管理	导入（转换）至华为HiLens的模型。
技能开发>技能管理	在控制台新建的技能资源。
数据管理（Beta）	不同空间的设备运行技能时所产生的日志。

创建工作空间后，所有工作空间可共享的资源，即无法实现隔离管理的资源如下：

表 4-2 共享资源

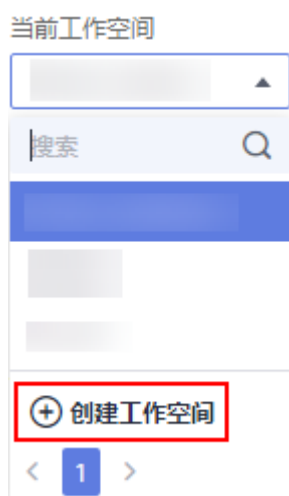
导航栏	共享资源
技能市场	技能市场可共享的技能以及使用当前账号购买的技能。
设备管理>产品管理	在华为HiLens新建的产品，即35xx芯片的摄像头设备。

导航栏	共享资源
技能开发>技能模板	华为HiLens预置的模板以及当前账号所收藏的模板。

创建工作空间

1. 登录华为HiLens管理控制台，在左侧导航栏中“当前工作空间”下方的下拉框中选择“创建工作空间”。
进入“创建工作空间”页面。

图 4-1 创建工作空间



2. 填写工作空间名称和描述，填完之后单击右下角的“立即创建”，跳到“工作空间”页面，可查看所创建的新工作空间。
填写工作空间描述时，不允许输入#~^\$%&*<>{}[]\|字符。

图 4-2 创建工作空间-0

* 工作空间名称

描述

0/255

分配工作空间

分配工作空间指把某工作空间的设备及其该设备下的技能、摄像头等资源挪到其他工作空间内。

此处以“test”作为设备资源所在的原工作空间，以“test2”作为设备资源分配的目标工作空间为例，介绍如何分配工作空间。

1. 登录华为HiLens管理控制台，在左侧导航栏中“当前工作空间”下方的下拉框中切换工作空间至“test”。

2. 在左侧导航栏中选择“设备管理>设备列表”，选择待分配的设备单击设备卡片。进入设备详情页。
3. 单击右上角的“分配工作空间”。弹出“分配工作空间”对话框。
4. 勾选工作空间“test2”，单击“确定”。
界面右上角弹出分配工作空间成功。您可以切换至工作空间“test2”，在“设备管理>设备列表”页面查看已分配的设备资源。

图 4-3 分配工作空间

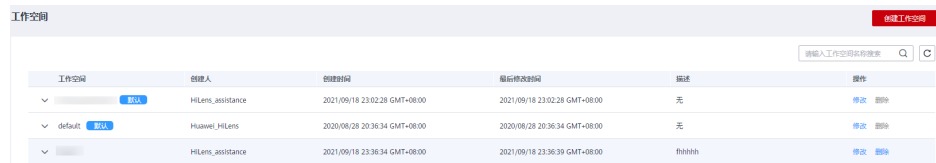


修改工作空间

基于已创建的工作空间，当前支持修改自定义工作空间的名称和描述，暂不支持修改默认工作空间的名称。

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“工作空间”。进入“工作空间”页面。

图 4-4 工作空间



2. 选择待修改的工作空间，单击“操作”列的“修改”。
3. 在对话框中重新编辑工作空间的“描述”，单击“确定”。

图 4-5 修改工作空间

修改

描述

不允许输入#~^\$%&+<>00V|字符。

0/255

确定

取消

删除工作空间

对于不再使用的工作空间，您可以删除工作空间以释放资源。默认工作空间不允许删除。

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“工作空间”。
进入“工作空间”页面。
2. 选择待修改的工作空间，单击“操作”列的“删除”。

图 4-6 删除工作空间

工作空间	创建人	创建时间	最后修改时间	描述	操作
<input type="checkbox"/> 默认	HiLens_assistance	2021/09/18 23:02:28 GMT+08:00	2021/09/18 23:02:28 GMT+08:00	无	修改 删除
<input type="checkbox"/> default	Huawei_HiLens	2020/08/28 20:36:34 GMT+08:00	2020/08/28 20:36:34 GMT+08:00	无	修改 删除
<input type="checkbox"/>	HiLens_assistance	2021/09/18 22:36:34 GMT+08:00	2021/09/18 22:36:39 GMT+08:00		修改 删除

3. 确认提示框中的内容，单击“确定”。

后续操作

创建不同的工作空间后，您可以在控制台左侧导航栏中，单击“当前工作空间”下方的下拉框，单击工作空间，进入所选工作空间内使用华为HiLens，可参考[使用流程简介](#)。

图 4-7 当前工作空间



5 注册 HiLens Kit

5.1 HiLens Kit 注册流程

HiLens Kit 简介

HiLens Kit是一款具备AI推理能力的多媒体终端设备，具有强大的计算性能、高清摄像头接入、体积小、接口丰富等特点。硬件集成了Atlas 200 AI加速模块（简称Atlas 200），方便用户快速简捷的接入并使用Ascend 310 AI处理器强大的处理能力，主要应用于AI开发者预研开发和视频监控领域。系统内置HiLens Framework，给运行的技能提供丰富的API接口。

HiLens Kit板载32GB eMMC，提供1个Micro SD卡槽位，可自行扩展Micro SD卡，最高支持速率SDR50，最大容量2TB，外接SD卡操作请见[使用SD卡](#)。其具体参数请参见[表5-1](#)。该产品的具体介绍见[华为云HiLens Kit用户指南>产品简介](#)。

图 5-1 HiLens Kit



表 5-1 HiLens Kit 参数说明

芯片	CPU	micro SD接口	摄像头	麦克风	网口	U S B	HD MI	Audio out	操作系统	Wi Fi
昇腾310 (8GB RAM)	Hi3559A (4GB)	1	200万像素, 720P	2	1	2	1	1	Linux	支持

参考文档

首次获取文档时，在企业技术支持网站（Support-E网站）注册账号并[注册产品](#)，输入HiLens Kit的产品序列号（SN），系统默认输入产品名称，完成产品注册申请，详情请参见[提升权限](#)。

SN码标注于HiLens Kit底部，为一串长达20的字符串，例如“21023XXXXXXXXXXXXXXXXXX”。

产品注册申请提交之后，如果是显示为“产品注册成功”，则可以直接执行下一步；如果提示“产品待审核”则需要等待审核成功之后执行下一步，一个工作日之内审核。

- [HiLens Kit用户指南](#)

介绍了Atlas 200 HiLens 套件（又称Atlas 200 HiLens Kit）的产品外观，逻辑结构和产品规格，指导用户对Atlas 200 HiLens Kit进行安装、连线、上电下电、基本配置等操作。

- 介绍了华为HiLens 智能边缘管理系统（Huawei HiLens Intelligent Edge System）的系统架构，并指导用户使用该管理系统对华为Atlas 200 HiLens Kit进行系统管理与维护等操作。

注册HiLens Kit方式

为了让HiLens Kit设备具备AI技能，需要将设备注册至HiLens控制台，才能在华为HiLens控制台上管理设备和技能等。

本章主要介绍如何登录端侧设备HiLens Kit并将设备注册至云侧华为HiLens控制台，在华为HiLens控制台上管理设备HiLens Kit。

- 目前只支持注册华为发布的HiLens Kit设备。
- 针对注册的HiLens Kit设备，只能安装并使用“适用于Ascend310芯片”的技能。

具体的操作流程和指导如下：

- [使用SSH注册设备](#)

常用操作

初始化配置：HiLens Kit出厂预装华为自研的Euler操作系统，用户无需安装操作系统。用户可通过浏览器登录华为HiLens 智能边缘管理系统，进行初始配置。

修改初始用户名和密码：为保证系统的安全性，初次登录时，请务必修改初始密码，并定期更新。

配置防火墙：为了防止除您允许的IP地址范围之外的用户去访问HiLens Kit智能边缘系统，您可以通过配置防火墙，防止HiLens Kit智能边缘系统受到其他攻击。

恢复出厂设置：在使用HiLens Kit过程中，如果系统被误破坏之后，无法恢复正常的使
用，需考虑恢复出厂设置。

使用SD卡：在HiLens Kit使用过程中，插入SD卡，存储技能数据。

使用 SSH 注册设备

使用SSH注册设备指通过SSH登录HiLens Kit系统，使用Linux命令注册设备。SSH登录设备系统支持操作HiLens Kit的系统文件等配置。

使用SSH注册设备流程如图5-2所示，详细操作指引请参见表5-2。

图 5-2 SSH 注册设备流程



表 5-2 SSH 注册设备流程说明

流程	说明	详细指导
连接PC和HiLens Kit	在注册HiLens Kit至控制台之前，您需要连接PC和HiLens Kit，并升级HiLens Kit系统固件版本至2.2.200.011。	连接PC和HiLens Kit
组网配置	HiLens Kit有两种组网方式，分为无线和有线两种方式连接路由器，您可以选择其中一种方式进行组网配置。 不能同时使用无线网络和有线网络连接同一个路由器，无线连接会自动删除默认网关，再次使用有线网络时需要配置默认网关。 推荐使用无线连接到路由器，以免更新设备IP后遗忘或丢失IP。	无线网络配置（SSH） 有线网络配置（SSH）
注册HiLens Kit	注册设备至控制台上，并在控制台上查看设备状态。	SSH注册HiLens Kit

5.2 使用 SSH 注册设备

5.2.1 连接 PC 和 HiLens Kit

如果需要操作HiLens Kit的系统文件等配置，可以使用SSH注册并登录设备，用Linux指令进行操作。下面以PuTTY工具为例，介绍在Windows 7环境下如何通过SSH登录HiLens Kit。

准备工作

- 确保您拥有至少一台HiLens Kit设备。
- 确保华为云账号没有欠费。
- 采用SSH注册设备，固件版本必须2.2.200.011及以上，如果低于此版本请先[升级HiLens Kit系统固件版本](#)。
系统固件版本查询方式：[登录华为HiLens智能边缘管理系统用户界面](#)，进入“维护>固件升级”页面，查看“当前版本号”。
- 确保获得如下数据：
 - 待连接设备的IP地址：
设备初始IP地址请参见中“管理网口初始IP地址”的“默认值”。
 - 登录待连接设备的用户名和密码：
设备的初始用户名和密码请参见中“初始用户名与密码”的“默认值”。
非首次登录，以修改后的账号名和密码登录，修改密码详情请见[修改初始用户名和密码](#)。
- 确保下载软件PuTTY.exe，此工具为第三方软件。

网线连接 PC 与 HiLens Kit

HiLens Kit后面板接口如[图5-3](#)和[表5-3](#)所示。

图 5-3 后面板接口



表 5-3 后面板接口说明



接口	说明
1	电源开关
2	电源接口

接口	说明
3	管理网口

1. 将DC 12V的电源适配器的端口插入HiLens Kit后面板的电源接口。
2. 打开HiLens Kit的电源开关（按住开关键1到2秒放开）。
3. 将网线的一端连接到设备的管理网口上，另一端连接到PC的以太网口上。

互通 PC 与 HiLens Kit 网络

设置PC机的IP地址、子网掩码或者路由，使PC机能和设备网络互通。

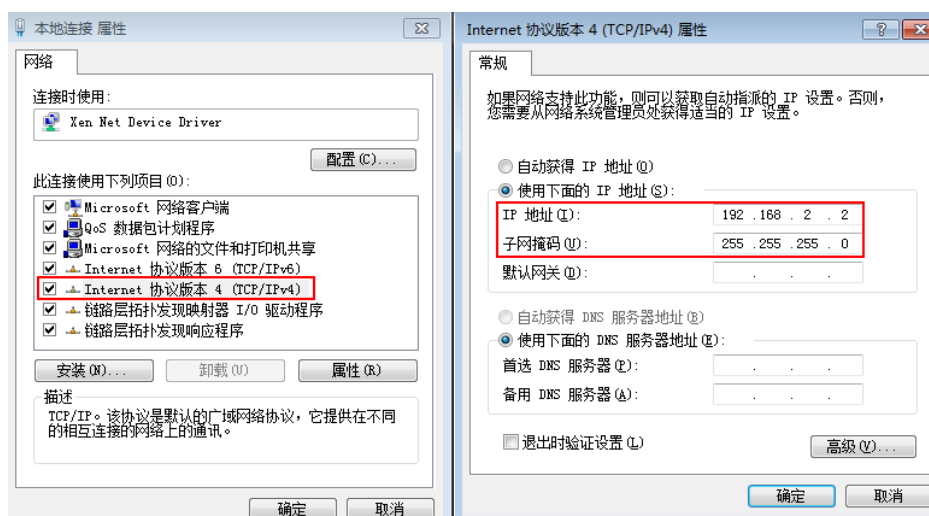
1. 单击PC右下角网络图标，单击“网络和Internet设置”。
2. 在“网络和Internet设置”页面，单击“更改适配器设置”，进入“网络连接”页面。
3. 设置HiLens Kit的网络连接，即HiLens Kit用网线连到PC后，在“网络连接”页面上会显示HiLens Kit对应的网络连接, 右键单击该网络连接（一般命名为“本地连接”），单击“属性”，弹出“属性”窗口。
4. 在“属性”设置窗口中，双击“Internet 协议版本4”，选择“使用下面的IP地址”，在右侧输入框中输入一个和设备**同一个网段**的IP（注：非端侧设备IP），单击“子网掩码”文本框，自动生成子网掩码，单击“确定”，完成网络属性修改。

设备的初始IP地址请参见[HiLens Kit 用户指南>默认数据](#)中“管理网口初始IP地址”的“默认值”。

说明

- 只有“本地连接属性>Internet协议版本4属性”中的IP和HiLens Kit设备IP在同一网段，才能从电脑SSH连接到HiLens Kit设备。**同一个网段**指本地连接IP的前三段地址要与设备IP一致。例如，设备IP是192.168.2.111，那么与设备IP为同一网段的IP，即“本地连接属性>Internet协议版本4属性”中的IP，可以是192.168.2.x，其中x是2-255中除111之外的整数。
- 如果设备IP已修改，此处“本地连接属性>Internet协议版本4属性”中的IP应填写与修改后的设备IP为同一网段的IP。修改IP的具体步骤请见[有线网络配置（SSH）](#)。

图 5-4 修改网络属性-6



首次注册HiLens Kit的用户，连接PC和HiLens Kit后，请[升级HiLens Kit系统固件版本](#)。

后续操作

使用SSH登录HiLens Kit设备，详情请见[连接PC和HiLens Kit](#)。

5.2.2 升级 HiLens Kit 系统固件版本

注册设备之前需要升级HiLens Kit系统固件版本至2.2.200.011（TR6）。升级HiLens Kit系统固件版本前请仔细阅读如下内容。

升级须知

- 系统固件版本低于2.2.200.011，可以进行升级固件。系统固件版本查询方式：[登录华为HiLens智能边缘管理系统用户界面](#)，进入“维护>固件升级”页面，查看“当前版本号”。
详情请参见[HiLens Kit升级指导书](#)。
- HiLens Kit系统固件升级后，版本会升级到2.2.200.011，同时HiLens_Device_Agent固件会升级到1.0.6（如果设备HiLens Kit处于使用状态，且当前HiLens_Device_Agent固件版本高于1.0.6，则会保留最新的HiLens_Device_Agent固件版本，不做更新）。
- 如果HiLens Kit为全新设备首次注册使用，我们推荐升级到最新固件版本，以获取更好的系统稳定性。
- 如果HiLens Kit已处于注册使用状态，我们建议您仔细阅读[升级风险](#)后，决定是否升级。

升级风险

升级前请仔细阅读以下风险，谨慎升级。

- HiLens Kit处于注册使用状态时，升级HiLens Kit系统固件版本前，请先联系接口人评估升级操作对业务连续性的风险。
- 升级HiLens Kit系统固件版本会重装整个系统目录，将导致系统目录下安装的软件会丢失。可采用如下方法确定系统目录的位置，用df -h命令可查看当前分区信息，“/dev/mmcblk0p2”对应根目录即是系统目录。此目录为软件默认安装目录，升级时该目录下安装的软件会丢失。其他非系统目录中的文件在升级过程中不受影响，如：“/dev/mmcblk0p*”对应的目录。
攻略：为了后面系统升级不受系统目录的影响，对于支持重定向安装的软件包，可以通过重定向安装到HiLens Kit设备系统“/opt”目录，规避以后系统升级，文件丢失问题。例如yum重定向安装软件包“XXX”，执行命令**yum --installroot=/opt install xxx**。对于不支持重定向安装的软件包，如yum安装、ROS安装涉及的rpm包，只能部署在系统目录，升级后会丢失。
- 升级HiLens Kit系统固件版本后可能会导致部分原有技能无法运行，请谨慎升级。

升级优势

- 支持端侧通过admin登录最长1个小时不掉线。
升级后，在admin用户模式下，通过命令**timeout**设置超时时间，最长1小时，例如设置超时时间为1小时，则执行命令**timeout 3600**。

- 首次登录智能边缘系统必须修改默认用户名、密码，防止使用默认用户名、密码带来的安全问题。
- 增加硬件设备和资源监测，当硬件异常或资源使用超过告警线，会上报告警提示您。
- 对第三方组件进行相关的安全升级。

前提条件

网线连接PC和设备，详细操作请参见[连接PC和HiLens Kit](#)。

升级操作

已使用过HiLens Kit的用户，升级前请仔细阅读[升级风险](#)。

请执行**两次**升级系统固件版本操作，操作步骤如下：

详细升级操作请参见[HiLens Kit升级指导书](#)。

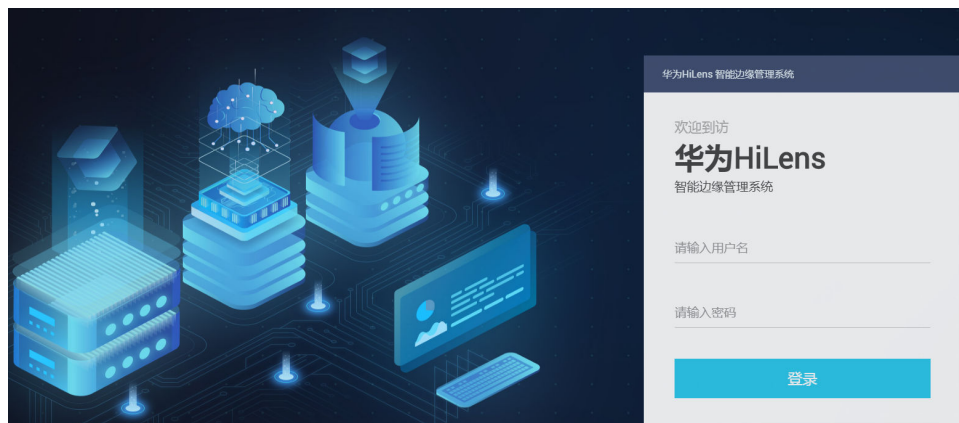
1. 首次获取升级包时，在企业技术支持网站（Support-E网站）注册账号并[注册产品](#)，输入HiLens Kit的产品序列号（SN），系统默认输入产品名称，完成产品注册申请，详情请参见[提升权限](#)。
SN码标注于HiLens Kit底部，为一串长达20的字符串，例如“21023XXXXXXXXXXXXXXXXXX”。
产品注册申请提交之后，如果是显示为“产品注册成功”，则可以直接执行下一步；如果提示“产品待审核”则需要等待审核成功之后执行下一步，一个工作日之内审核。
2. 登录[华为企业业务网站](#)，在“技术支持>产品支持>昇腾计算>智能边缘硬件”产品列表中选择[A200-3000HiLens](#)。
如果产品选择错误，将会导致SN不通过，无法顺利执行升级操作。
3. 单击“软件”，在软件版本列表中选择目标版本“A200-3000HiLens-FWV2.2.200.011.hpm”，下载升级包至本地PC。
4. 登录华为HiLens智能边缘管理系统，在本地PC中打开浏览器。在地址栏中输入华为HiLens智能边缘管理系统的地址，地址格式为“[https://华为HiLens智能边缘管理系统的访问IP地址](#)”（默认IP为192.168.2.111）。按“Enter”键。


说明

如果浏览器显示“此网站的安全证书有问题”，请单击“继续浏览此网站(不推荐)”。

5. 在系统登录界面中，配置登录信息。
在“用户名”和“密码”输入框中输入登录的用户名和密码。
初始用户名和密码请见。

图 5-5 登录华为 HiLens 智能边缘管理系统



- 在主菜单中选择“维护>固件升级>系统固件升级”。
进入“系统固件升级”页面。
- 单击“升级文件”后的 ，选择文件，即步骤2中下载的升级包。
界面提示“已添加文件”。

说明

上传过程中，请勿切换或者关闭当前页面，否则会导致上传失败。

- 单击“升级”，在弹出的提示框中勾选“升级完成后系统自动重启生效”。
如果不勾选该选项，则需手动重启生效。
- 单击“确定”。
可在页面查看升级包版本号、升级进度等信息。
- 等待10分钟左右，提示升级成功。
如果首次执行升级操作，重新登录华为HiLens智能边缘管理系统，请重复执行步骤3-步骤8。

说明

为确保设备注册成功，在执行第二次升级操作时，请使用同一个升级包。
如果已执行升级操作两次，则已完成HiLens Kit固件版本升级。

5.2.3 SSH 登录 HiLens Kit 设备

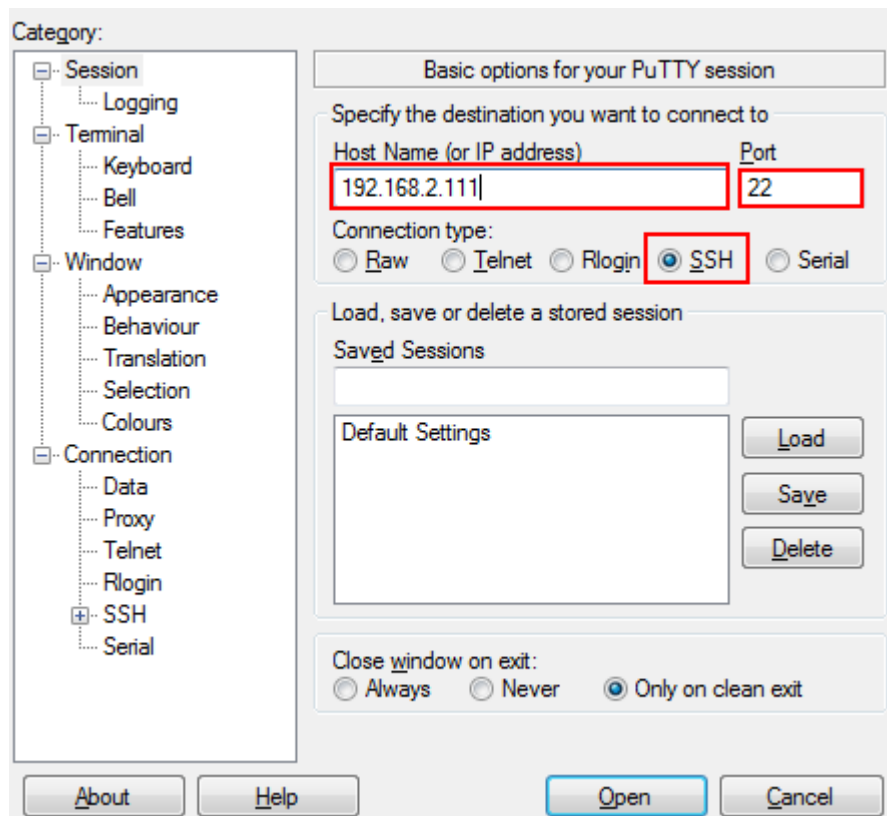
前提条件

已连接PC和HiLens Kit，详情请见[连接PC和HiLens Kit](#)。

操作步骤

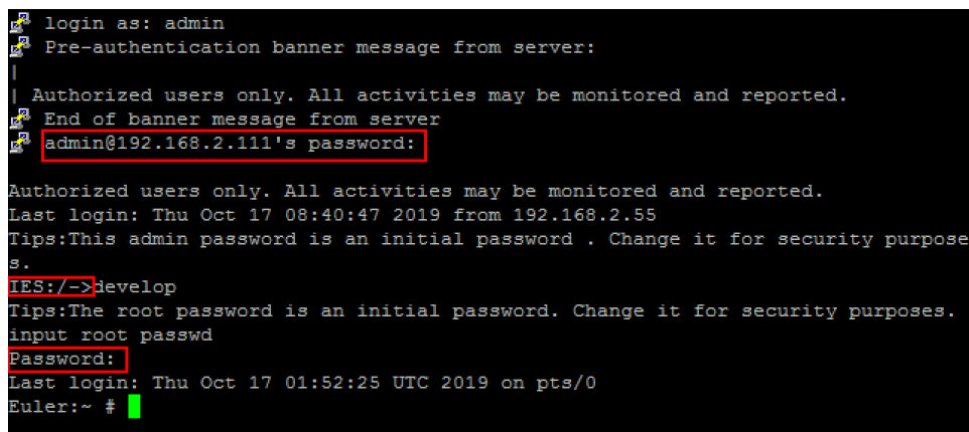
- 使用SSH远程连接HiLens Kit设备。
 - 运行PuTTY，单击“Session”，在“Host Name(or IP address)”的输入框中输入设备IP，在“Port”输入框中输入端口号。以设备IP是192.168.2.111，设备端口号是22为例，如[图5-6](#)所示。

图 5-6 使用 PuTTY 登录 HiLens Kit



- b. 单击“open”。
登录端侧设备。
2. 使用SSH远程登录设备系统。
在“admin@192.168.2.111 's password”提示语后输入默认账户“admin”的密码，首次登录默认密码请参见。
3. 使用SSH远程登录开发者命令行界面。
 - a. 在“IES: /->”提示语后执行命令**develop**。
 - b. 在“Password”提示语后输入root密码，首次登录默认密码请参见。

图 5-7 SSH 登录设备



4. 使用SSH远程修改设备时间:

- a. 调整时区，与时区一致，执行命令
timedatectl set-timezone
- b. 修改设备时间，以当前时间为2019年10月17日19时19分19秒为例。执行命令
date -s "2019-10-17 19:19:19"
hwclock -w
reboot
重启设备。

说明

设备默认注册至区域北京四，需要把设备注册至新加坡，[切换设备注册区域](#)。

切换设备注册区域至新加坡（可选）

设备默认注册至区域北京四，需要把设备注册至新加坡，请修改设备中配置文件的相关内容。详细步骤如下：

1. 在PuTTY里进入文件
vi /home/hilens/hda/etc/hda.conf
2. 修改设备注册区域：在注释语“#选择注册到华为云的区域，默认为北京4”下方，把**region=cn-north-4**修改为
region = ap-southeast-3
并保存。

说明

必须使用国际站账号才能注册设备到新加坡区域，详情请见[中国站与国际站的差异](#)。如果没有国际站账号请先在华为云国际站站点注册账号，详情请见[注册华为帐号并开通华为云](#)。

3. 重启设备，即执行
reboot

后续操作

组网配置：HiLens Kit有两种组网方式，分为无线和有线两种方式连接路由器，您可以选择其中一种方式进行组网配置。

两种不同组网方案的连接方式的操作指导如下：

- [无线网络配置（SSH）](#)
- [有线网络配置（SSH）](#)

不能同时使用无线网络和有线网络连接同一个路由器，无线连接会自动删除默认网关，再次使用有线网络时需要配置默认网关。**推荐使用无线连接到路由器，以免更新设备IP后遗忘或丢失IP。**

5.2.4 无线网络配置（SSH）

HiLens Kit有两种组网方式，分为无线和有线两种方式连接路由器，本章介绍无线网络配置方式。

📖 说明

不能同时使用无线网络和有线网络连接同一个路由器，无线连接会自动删除默认网关，再次使用有线网络时需要配置默认网关。**推荐使用无线连接到路由器，以免更新设备IP后遗忘或丢失IP。**如果选择有线网络配置，请参见[有线网络配置（SSH）](#)。

配置要求

针对使用无线网络连接路由器的方式，需要输入无线网络密码，成功连接无线网络。

当前HiLens Kit仅支持**2.4G**频段的无线网络和常规的**加密类型**无线网络，且**无线网络名称**不包含英文的单引号和双引号，长度8-63个字符。

- 支持的2.4G频段无线网络所使用的协议为IEEE802.11n\IEEE802.11g\IEEE802.11b。
- 支持的无线网络加密类型有WEP、WPA-PSK/WPA2-PSK和AES。
- 暂不支持需要验证的无线网络。
- 暂不支持TKIP加密。
- 路由器网关不能使用xxx.xxx.0.xxx网段，比如**192.168.0.1**，也不要和HiLens Kit网口在同一网段，例如，设备IP是**192.168.2.111**，那么路由器的网关就不能设置成**192.168.2.xxx**。

前提条件

- 网线连接PC和设备，详细操作请参见[连接PC和HiLens Kit](#)。
- 升级HiLens Kit系统固件版本，详情请参见[升级HiLens Kit系统固件版本](#)。

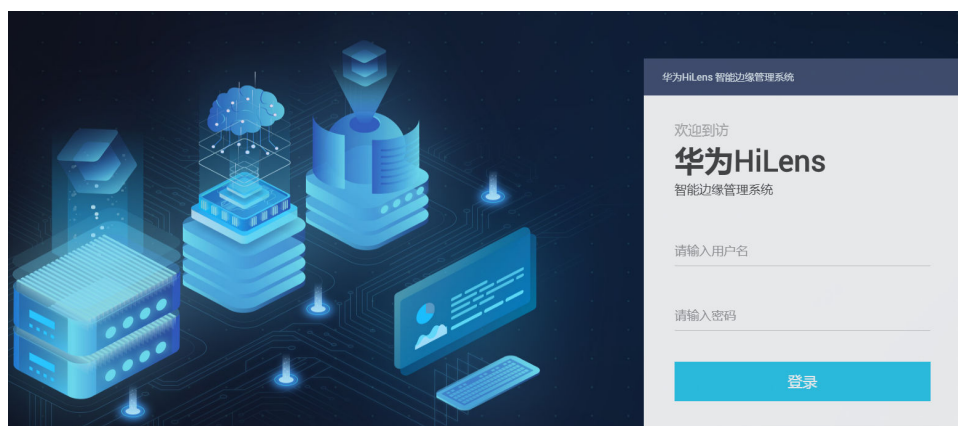
操作步骤

1. 登录华为HiLens智能边缘管理系统，在本地PC中打开浏览器。在地址栏中输入华为HiLens智能边缘管理系统的地址，地址格式为“**https://华为HiLens智能边缘管理系统的访问IP地址**”（默认IP为192.168.2.111）。按“Enter”键。

📖 说明

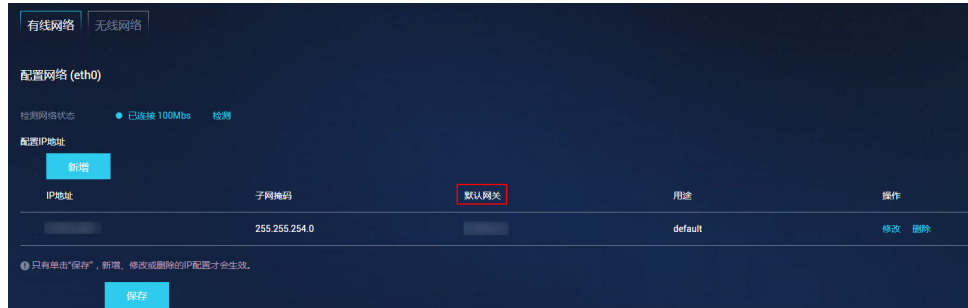
- 如果浏览器显示“此网站的安全证书有问题”，请单击“继续浏览此网站(不推荐)”。
2. 在系统登录界面中，配置登录信息。
在“用户名”和“密码”输入框中输入登录的用户名和密码。
初始用户名和密码请见。

图 5-8 登录华为 HiLens 智能边缘管理系统-2



3. 在主菜单中选择“管理>网络>有线网络”。
进入“有线网络”配置页面。
4. 检查“配置IP地址”区域的“IP地址”是否存在“默认网关”，如图5-9所示。

图 5-9 检查默认网关






- 如果不存在默认网关，请执行下一步。
 - 如果存在默认网关，请执行以下操作：
单击操作栏“修改”，在弹出的“修改IP地址”对话框中，删除“默认网关”文本框中的已有值，单击“确定”。
执行下一步。
5. 在主菜单中选择“管理>网络>无线网络”。
进入“无线网络”配置页面。
 6. 单击, 打开WIFI开关。
搜索附近热点，可连接附近的无线网络。
 7. 单击, 可刷新热点信息。
 8. 单击。
弹出信息提示框。
 9. 在“WIFI密码”文本框中输入WIFI密码，图5-10所示。单击“确定”。
页面右上角弹出“连接成功”，完成无线网络连接路由器。

图 5-10 无线连接路由器



查看无线网络 IP

配置无线网络后，您可以在华为HiLens智能边缘管理系统获取无线网络的IP地址。

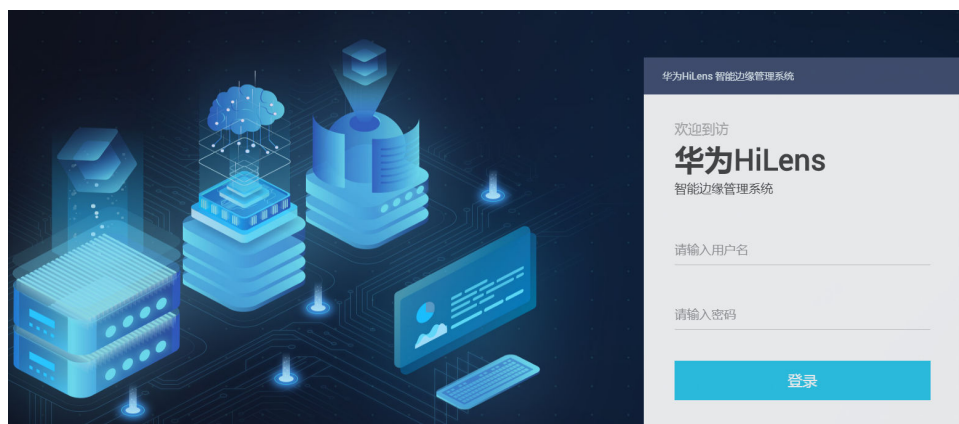
1. 登录华为HiLens智能边缘管理系统，在本地PC中打开浏览器。在地址栏中输入华为HiLens智能边缘管理系统的地址，地址格式为“**https://**华为HiLens智能边缘管理系统的访问IP地址”（默认IP为192.168.2.111）。按“Enter”键。

说明

如果浏览器显示“此网站的安全证书有问题”，请单击“继续浏览此网站(不推荐)”。

2. 在系统登录界面中，配置登录信息。
在“用户名”和“密码”输入框中输入登录的用户名和密码。
初始用户名和密码请见。

图 5-11 登录华为 HiLens 智能边缘管理系统-3



3. 在主菜单中选择“管理>网络>无线网络”。
进入“无线网络”配置页面。

4. 在您所连接的无线网络卡片右上角，单击“连接信息”。
弹出“查看WIFI信息”窗口。
“IP地址”即无线网络的IP地址。

图 5-12 查看无线 IP



后续操作

注册HiLens Kit，详情请见[SSH注册HiLens Kit](#)。

5.2.5 有线网络配置（SSH）

HiLens Kit有两种组网方式，分为无线和有线两种方式连接路由器，本章介绍有线网络配置方式。

说明

不能同时使用无线网络和有线网络连接同一个路由器，无线连接会自动删除默认网关，再次使用有线网络时需要配置默认网关。**推荐使用无线连接到路由器，以免更新设备IP后遗忘或丢失IP。**
无线网络配置请参见[无线网络配置（SSH）](#)。

配置说明

针对使用有线网络直连路由器的方式，需要修改设备IP，使设备IP与路由器IP在同一网段。

同一个网段指设备连接IP的前三段地址要与路由器IP一致。例如，设备IP是192.168.2.111，那么路由器IP可以是192.168.2.x，其中x是2-255中除111之外的整数。

前提条件

- 网线连接PC和设备，详细操作请参见[连接PC和HiLens Kit](#)。
- 升级HiLens Kit系统固件版本，详情请参见[升级HiLens Kit系统固件版本](#)。

操作步骤

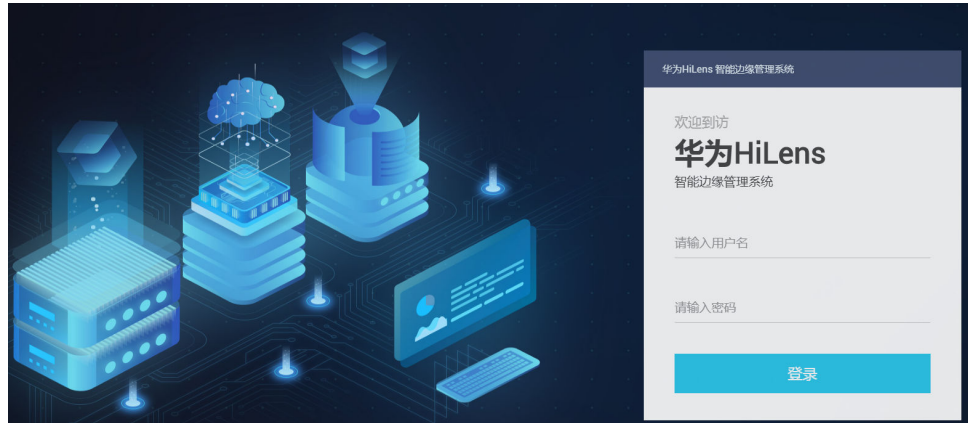
1. 登录华为HiLens智能边缘管理系统，在本地PC中打开浏览器。在地址栏中输入华为HiLens智能边缘管理系统的地址，地址格式为“[https://华为HiLens智能边缘管理系统的访问IP地址](#)”（默认IP为192.168.2.111）。按“Enter”键。

说明

如果浏览器显示“此网站的安全证书有问题”，请单击“继续浏览此网站(不推荐)”。

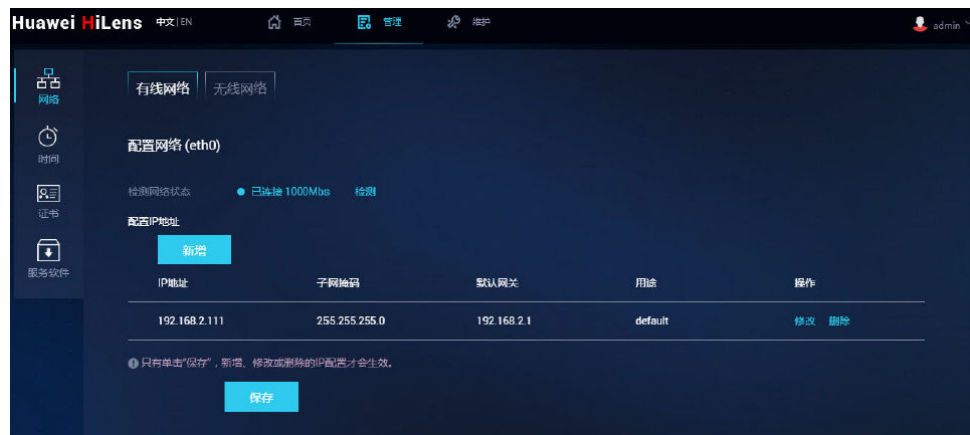
2. 在系统登录界面中，配置登录信息。
在“用户名”和“密码”输入框中输入登录的用户名和密码。
初始用户名和密码请见。

图 5-13 登录华为 HiLens 智能边缘管理系统-7



3. 在主菜单中选择“管理>网络>有线网络”。
进入“有线网络”配置页面。

图 5-14 有线连接路由器-8



4. 单击检测网络状态后的“检测”，检查网络是否连接。
5. 修改IP地址。
 - a. 单击待修改IP地址后的“修改”。
 - b. 按表5-4修改“IP地址”。本示例使用路由器IP是“192.168.2.1”，如图5-15所示，单击“确定”，完成修改设备IP。
 - c. 单击“保存”，重启系统，使IP配置生效。

表 5-4 修改 IP 参数

参数	说明
用途	IP地址的用途。 支持输入英文、数字和下划线字符，字符串长度1~32个字符。
IP地址	需要修改的IPv4地址。 此处IP地址应和路由器IP在同一网段的IP地址。
默认网关	该IP地址对应的默认网关。 说明 请确保默认网关全局唯一。

图 5-15 修改 IP 地址-9

6. 拔出PC侧网线，断开设备与PC的网线连接，然后网线连接设备与路由器。
7. 连接完成后，SSH登录设备。执行命令检查是否连接成功。

ping 8.8.8.8

或者

ping www.huaweicloud.com

如果设备连接成功，其提示信息将显示如下类似信息。

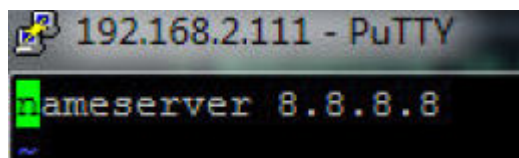
图 5-16 有线连接提示信息

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=50 time=86.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=50 time=172 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=50 time=125 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 86.476/127.914/171.826/34.887 ms
```

如果设备连接失败，请执行下一步。

8. 使用SSH远程配置DNS。
 - a. 在PuTTY里进入文件
vi /etc/resolv.conf
 - b. 删掉原有内容，并修改为
nameserver 8.8.8.8
 - c. 如图5-17所示，并保存退出。重新执行上一步检查设备是否连接成功。

图 5-17 配置 DNS 内容



后续操作

注册HiLens Kit，详情请见[SSH注册HiLens Kit](#)。

5.2.6 SSH 注册 HiLens Kit

前提条件

- 网线连接PC和设备，详细操作请参见[连接PC和HiLens Kit](#)。
- 升级HiLens Kit系统固件版本，详情请参见[升级HiLens Kit系统固件版本](#)。
- 选择如下方式中的一种组网配置：
 - [无线网络配置（SSH）](#)
 - [有线网络配置（SSH）](#)

不能同时使用无线网络和有线网络连接同一个路由器，无线连接会自动删除默认网关，再次使用有线网络时需要配置默认网关。**推荐使用无线连接到路由器，以免更新设备IP后遗忘或丢失IP。**

操作步骤

1. 使用SSH远程注册HiLens Kit设备至华为HiLens控制台，执行命令
hdctl register -u用户名 -d账号名 -n设备名
按回车弹出“password”提示语。
用户名指IAM用户名，**账号名**指华为云账号名，相关概念请参见[IAM基本概念](#)。
如果没有IAM账户时，“账号名”与“用户名”一致。

📖 说明

- 注册HiLens Kit所使用的账号名**必须是华为云账号，不是华为账号**。
华为帐号和华为云帐号不同，如果您暂未注册华为账号，建议您先注册华为账号并开通华为云，请参考[注册华为帐号并开通华为云](#)。
 如果您使用的账号是华为账号，请使用此账号创建子账号，详情请见[创建IAM用户](#)，并给子账号添加admin权限。
- 注册设备到新加坡区域时，必须使用国际站账号的**用户名和账号名**，详情请见[中国站与国际站的差异](#)。如果没有国际站账号，请先在华为云国际站站点注册账号，详情请见[注册华为帐号并开通华为云](#)。

设备名请自己定义。

注册HiLens Kit所使用的用户名、账号名、设备名支持英文字母、数字和下划线，不支持以数字开头的名称以及只有数字的名称。

- 在“password”提示语后输入华为云账号的密码，按回车完成设备注册。
 如果IAM用户和华为云账号不一致，在“password”提示语后输入IAM用户密码。
 如果有报错提示，无法注册，请将设备[恢复出厂设置](#)，重新注册设备。

📖 说明

- IAM用户禁止将设备注册到子项目，相关基本概念请参见[IAM基本概念](#)。
 - 设备注册完成后，您可登录华为HiLens管理控制台，在“设备管理>设备列表”中查看设备状态。短时间内处于离线状态，请您耐心等待。
- 登录华为HiLens管理控制台，单击左侧导航栏“设备管理>设备列表”，在“设备列表”查看设备状态，如[图5-18](#)所示。

图 5-18 设备状态



- 等待3分钟，如果设备处于在线状态，则成功重启设备。
- 等待3分钟，如果设备仍处于离线状态，则需要检查设备时间是否与当前时间一致，详情请参见设备离线。

在华为 HiLens 管理控制台查看注册后的设备

登录华为HiLens管理控制台，请在华为HiLens管理页面左上角切换至“新加坡”，在管理控制台左侧菜单栏单击“设备管理 > 设备列表”，默认设备列表展现所有设备，查看到您所注册的设备列表，且设备状态处于“在线”状态，则说明您的设备成功注册。

如果注册失败或设备状态处于离线状态，请参见[HiLens Kit注册失败](#)排查原因。

图 5-19 设备注册状态



须知

查看设备注册状态时：

- 注册设备后，请升级HiLens_Device_Agent固件版本至1.0.7及其以上版本，详细操作请参见[升级HiLens_Device_Agent固件版本](#)。

5.3 组网配置

5.3.1 组网配置方式

HiLens Kit分无线和有线两种方式连接路由器，您可以选择其中一种方式进行组网配置，也可以同时使用无线和有线的连接外网和公司、学校等内部网络。

- 场景一：使用无线网络配置
[无线网络配置 \(SSH\)](#)
- 场景二：使用有线网络配置
[有线网络配置 \(SSH\)](#)
- 场景三：使用无线网络连接外网，使用有线网络连接公司、学校等内部网络
[无线网络 \(外网\) & 有线网络 \(内网\)](#)

📖 说明

不建议同时使用无线网络和有线网络连接同一个路由器，推荐使用无线连接到路由器，以免更新设备IP后遗忘或丢失IP。

5.3.2 无线网络（外网）&有线网络（内网）

本节主要介绍如何使用无线网络（WIFI）连接外网，同时使用有线网络连接公司、学校等内部网络的操作方式。

方式一：先配置无线网络，再配置有线网络，然后添加被删除的无线IP默认路由表项。

方式二：先配置有线网络，再配置无线网络，然后添加被删除的有线IP默认路由表项。

本节以方式一详细介绍HiLens Kit同时使用无线和有线进行组网配置的过程。

准备工作

1. 网线连接本地PC和HiLens Kit。
2. 在浏览器输入“https://XXX.XXX.XXX.XXX”，并按“Enter”打开智能边缘系统登录界面（HiLens IES Web）。
其中“XXX.XXX.XXX.XXX”为设备默认IP，默认参数请见[默认参数](#)。
3. 在“用户名”和“密码”输入框中输入登录的用户名和密码，默认参数请见[默认参数](#)。
4. 使用默认IP，SSH登录HiLens Kit，进入管理员（develop）权限命令行界面，详情请见[SSH登录HiLens Kit设备](#)。

如果公司、学校等内部网络分配IP要求与接入设备MAC地址绑定，请固化HiLens Kit有线网口(eth0) MAC地址。

- a. 在HiLens Kit管理员（develop）权限命令行界面输入命令**ifconfig**，查看HiLens Kit设备有线网口的MAC地址。

如[图5-20](#)所示，红框所标识的内容就是HiLens Kit有线网口“eth0”的MAC地址。

图 5-20 查看 MAC 地址

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.111 netmask 255.255.255.0 broadcast 192.168.2.255
    inet6 fe80::aeb3:b5ff:fe2e:8763 prefixlen 64 scopeid 0x20<link>
    ether ac:b3:b5:2e:87:63 txqueuelen 1000 (Ethernet)
    RX packets 452 bytes 44420 (43.3 KiB)
    RX errors 0 dropped 4 overruns 0 frame 0
    TX packets 524 bytes 36961 (36.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 40
```

- b. 在管理员（develop）权限命令行界面打开文件“vi /etc/rc.d/rc.local”，追加如下内容：
Ifconfig eth0 down
Ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx（“xx:xx:xx:xx:xx:xx”为MAC地址）
Ifconfig eth0 up
- c. 重启设备使修改操作生效。

步骤一：配置无线网络

使用无线网络连接外网，动态获取无线网络的IP地址。针对使用无线网络连接路由器的方式，需要输入无线网络密码，成功连接无线网络。

- **操作步骤**
无线网络的配置要求及详细步骤请见[无线网络配置（SSH）](#)。
- **网络验证**
 - a. 在命令行界面执行命令**ifconfig**，查看HiLens Kit的IP信息。
以无线网络IP为“192.168.137.26”为例，“wlan0”表示无线网口。

图 5-21 设备 IP-1

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.137.26 netmask 255.255.255.0 broadcast 192.168.137.255
    inet6 fe80::20f:ff:fe:cf:c2:df prefixlen 64 scopeid 0x20<link>
    ether 00:0f:00:cf:c2:df txqueuelen 1000 (Ethernet)
    RX packets 260 bytes 30626 (29.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 157 bytes 18923 (18.4 KiB)
```

- b. 执行命令**route**，查看路由表项信息。
以无线网络IP为“192.168.137.26”，所分配的默认网关为“192.168.137.1”为例。

图 5-22 路由表项信息-1

```
Euler:~ # route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 192.168.137.1 0.0.0.0 UG 0 0 0 wlan0
link-local 0.0.0.0 255.255.0.0 U 1002 0 0 eth0
172.17.0.0 0.0.0.0 255.255.0.0 U 0 0 0 docker0
192.168.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
192.168.137.0 0.0.0.0 255.255.255.0 U 0 0 0 wlan0
Euler:~ #
```

- c. 执行命令检查无线网络是否连接成功。
ping 8.8.8.8
如果设备连接成功，其提示信息将显示如下类似信息。

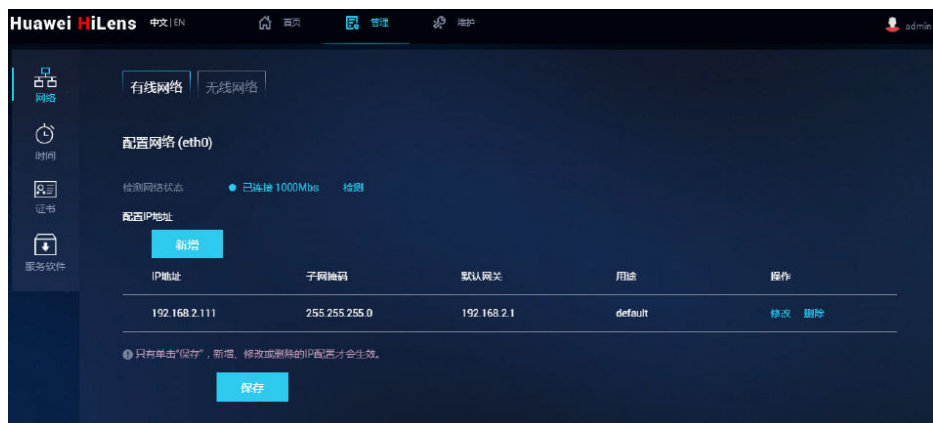
图 5-23 有线连接提示信息-1

```
PING www. (103.235.46.39) 56(84) bytes of data:
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=3 ttl=43 time=80.3 ms
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=13 ttl=43 time=528 ms
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=15 ttl=43 time=199 ms
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=16 ttl=43 time=86.9 ms
64 bytes from 103.235.46.39 (103.235.46.39): icmp_seq=17 ttl=43 time=166 ms
```

步骤二：配置有线网络

- **操作步骤**
使用有线网络连接公司、学校等内部网络，需要在有线网口新增内网IP。如果需要域名解析，请配置对应的DNS。
 - a. 在浏览器输入“https://XXX.XXX.XXX.XXX”，并按“Enter”打开智能边缘系统登录界面（HiLens IES Web）。
其中“XXX.XXX.XXX.XXX”为设备默认IP，默认参数请见[默认参数](#)。
 - b. 在“用户名”和“密码”输入框中输入登录的用户名和密码，默认参数请见[默认参数](#)。
 - c. 在主菜单中选择“管理>网络>有线网络”。
进入“有线网络”配置页面。

图 5-24 有线网络配置



- d. 在配置IP地址下方单击“新增”，按表5-5在新增IP对话框中填写内网IP信息，并单击“确定”。
- e. 在有线网络页面的“配置IP地址”下方单击“保存”。

表 5-5 新增 IP 参数

参数	说明
用途	IP地址的用途。 支持输入英文、数字和下划线字符，字符串长度1~32个字符。
IP地址	需要修改的IPv4地址。 此处IP地址应和路由器IP在同一网段的IP地址。
默认网关	该IP地址对应的默认网关。 说明 请确保默认网关全局唯一。

图 5-25 添加 IP 地址



新增IP地址

用途 wired_network

IP地址 10 ·

子网掩码 255 ·

默认网关 10 ·

VLAN ID

若网络配置中存在相同网段IP, 请确保配置的IP地址正确, 且与网络上的其它IP无冲突, 否则将导致网络异常。

确定 取消

- f. 配置内部网络对应的DNS地址。根据是否升级HiLens Kit系统固件版本，配置DNS的方式也不同。

■ 已升级HiLens Kit系统固件版本

已升级HiLens Kit系统固件版本至2.2.200.011，详情请见[升级HiLens Kit系统固件版本](#)，配置DNS方式如下。

- 1) 在智能边缘系统的“管理>网络>有线网络”页面。单击“配置DNS地址”的“新增”。
弹出添加DNS地址对话框。

图 5-26 添加 DNS



添加DNS地址

10 ·

确定 取消

- 2) 在对话框中填写公司或学校内部网络的DNS地址，单击“确定”。
 - 3) 在有线网络页面的“配置DNS地址”下方单击“保存”。
 - **未升级HiLens Kit系统固件版本**
使用SSH远程配置DNS。
在PuTTY里进入文件“vi /etc/resolv.conf”，新增DNS：
nameserver x.x.x.x（x.x.x.x为新增的DNS地址）
保存并退出。
- **网络验证**
 - a. 使用默认IP，SSH登录HiLens Kit，详情请见[SSH登录HiLens Kit设备](#)。
 - b. 执行命令**ifconfig**，查看HiLens Kit的IP信息。
以有线网络IP为“10.78.14.58”为例，“eth0”表示有线网口，“eth0:1”为新增IP地址。

图 5-27 设备 IP

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.111 netmask 255.255.255.0 broadcast 192.168.2.255
    inet6 fe80::4f5:81ff:fe49:e92b prefixlen 64 scopeid 0x20<link>
    ether 06:f5:81:49:e9:2b txqueuelen 1000 (Ethernet)
    RX packets 5864 bytes 1320409 (1.2 MiB)
    RX errors 0 dropped 2 overruns 0 frame 0
    TX packets 4706 bytes 16730604 (15.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 40

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.78.14.58 netmask 255.255.254.0 broadcast 10.78.15.255
    ether 06:f5:81:49:e9:2b txqueuelen 1000 (Ethernet)
    device interrupt 40
```

- c. 执行命令**route**，查看路由表项信息。
以有线网络IP为“10.78.14.58”，，所分配的默认网关为“192.168.137.1”为例，会发现**步骤一：配置无线网络**分配的IP对应默认路由信息没有了，导致此时无法访问外网。

图 5-28 路由表项信息

```
Euler:~ # route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 10.78.14.1 0.0.0.0 UG 0 0 0 eth0
10.78.14.0 0.0.0.0 255.255.254.0 U 0 0 0 eth0
link-local 0.0.0.0 255.255.0.0 U 1002 0 0 eth0
172.17.0.0 0.0.0.0 255.255.0.0 U 0 0 0 docker0
192.168.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
192.168.137.0 0.0.0.0 255.255.255.0 U 0 0 0 wlan0
```

步骤三：添加路由表项

手动添加**步骤一：配置无线网络**时分配的IP所对应的默认路由表项。

- **操作步骤**
 - a. 使用默认IP，SSH登录HiLens Kit，进入管理员（develop）权限命令行界面，详情请见[SSH登录HiLens Kit设备](#)。
 - b. 在命令行界面执行命令，添加默认路由表项。
以**步骤一：配置无线网络**分配的IP所对应的默认网关是“192.168.137.1”为例。

route add default gw 192.168.137.1

- c. 拔出PC侧网线，断开PC和HiLens Kit的网线，网线连接HiLens Kit和内部网络。

- **网络验证**

- a. 在内网PC上使用HiLens Kit新配置内网IP（本例中是“10.78.14.58”）SSH登录HiLens Kit，详情请见[SSH登录HiLens Kit设备](#)。
- b. 依次执行命令**ping**检查是否成功连接有线网络（外网）和无线网络（内网）。

如果设备连接成功，其提示信息将显示如下类似信息。

图 5-29 有线网络连接提示信息

```
Euler:~ # ping 10.78.15.147
PING 10.78.15.147 (10.78.15.147) 56(84) bytes of data.
64 bytes from 10.78.15.147: icmp_seq=1 ttl=127 time=0.897 ms
64 bytes from 10.78.15.147: icmp_seq=2 ttl=127 time=0.891 ms
64 bytes from 10.78.15.147: icmp_seq=3 ttl=127 time=0.907 ms
```

图 5-30 无线网络连接提示信息

```
Euler:~ # ping www. .com
PING www.wshifen.com (104.193.88.77) 56(84) bytes of data.
64 bytes from 104.193.88.77 (104.193.88.77): icmp_seq=1 ttl=44 time=220 ms
64 bytes from 104.193.88.77 (104.193.88.77): icmp_seq=2 ttl=44 time=224 ms
```

5.4 配置防火墙

为了防止除您允许的IP地址范围之外的用户去访问HiLens Kit智能边缘系统，建议您通过SSH远程配置防火墙，防止HiLens Kit智能边缘系统受到其他攻击。

配置方式

- **配置硬件防火墙**
如果存在硬件防火墙，请联系运维人员配置允许访问HiLens Kit设备IP地址的硬件防火墙。
- **配置软件防火墙**
如果不存在硬件防火墙，请配置软件防火墙，具体步骤请见[配置软件防火墙](#)。

配置软件防火墙

1. 网线连接电脑和设备，使用SSH登录HiLens Kit设备，详细操作指导请参见[连接PC和HiLens Kit](#)。
2. 在PuTTY里执行命令
iptables -I INPUT -p tcp --dport 443 -j DROP
禁止所有用户通过443端口，访问HiLens Kit智能边缘系统。
3. 根据自身所需，配置允许通过443端口访问HiLens Kit智能边缘系统的IP地址。
 - 配置HiLens Kit设备IP访问HiLens Kit智能边缘系统，以设备IP是192.168.2.111为例，在PuTTY里执行命令
iptables -I INPUT -s 192.168.2.111 -p tcp --dport 443 -j ACCEPT

- 配置单个IP地址访问HiLens Kit智能边缘系统，以允许访问的IP地址是10.61.120.127为例，在PuTTY里执行命令
iptables -I INPUT -s 10.61.120.127 -p tcp --dport 443 -j ACCEPT
 - 配置某一网段IP访问HiLens Kit智能边缘系统，以网段IP地址是10.61.120.*（*为0-255中任意整数）为例，在PuTTY里执行命令
iptables -I INPUT -s 10.61.120.0/24 -p tcp --dport 443 -j ACCEPT
4. 在PuTTY里进入目录/home/data/user，将步骤2和步骤3的配置命令写入user_init.sh文件，防止重启HiLens Kit设备后步骤2和步骤3的配置失效。
如果目录/home/data/user下没有user_init.sh文件，请重新创建user_init.sh文件。

5.5 使用 SD 卡

HiLens Kit是端云协同多模态AI开发套件，使用过程中会产生视频、图像等数据，通常情况下通过接口将数据存储到OBS。如果需要将数据存储在本地，由于HiLens Kit自带的硬盘空间有限，需要外插SD卡来存储。

操作步骤

1. SD卡插入HiLens Kit后面板接口“Micro SD”。
2. SSH登录HiLens Kit，详情请见[SSH登录HiLens Kit设备](#)。
3. 查找SD卡位置，执行命令
ls -al /dev/mmcblk*
执行结果为
dev/mmcblk1
4. 保存和备份SD卡已存储内容，格式化SD卡。
mkfs.ext4 /dev/mmcblk1
请在EulerOS提示符下执行。
如果SD卡已格式化，请忽略此步骤。
5. 创建SD路径，执行命令
mkdir /mnt/sd
设备挂载到该路径下“mount /dev/mmcblk1/mnt/sd”。
6. 查询文件系统，执行命令df。
可执行以下命令，对本地文件test.txt进行读写测试。
 - 写命令：**cp test.txt /mnt/sd**
 - 读命令：**cp /mnt/sd/test.txt ./**

说明

SD卡使用完并拔出之前，请先卸载设备挂载的分区，执行命令
umount /mnt/sd
防止有文件损坏。

6 管理设备

6.1 设备管理简介

平台目前支持HiLens Kit、Atlas 500智能小站的注册、管理和注销。针对使用海思35xx系列芯片的设备，其设备厂商可以利用HiLens管理控制台进行批量管理，并为其产品赋予AI能力，详细操作指导请参考[产品简介](#)。下文所有操作，仅针对HiLens Kit设备。

在进行设备管理之前，您需要将端侧设备注册至华为HiLens中，其相应的说明和连接指导请参见[注册HiLens Kit](#)。

设备管理相关操作及说明

- **管理设备上的技能**：可以在已注册设备上安装技能、配置运行时参数、启动技能、停止技能或卸载技能等操作。
- **查看设备信息**：您可以查看设备的基本信息或升级状态，了解更多设备相关信息。
- **查看设备日志**：您可以在使用HiLens技能后，查看设备agent日志和技能日志。
- **升级HiLens_Device_Agent固件版本**：当设备厂商有新的固件版本时，您可以选择升级固件版本。
- **配置摄像头**：为已注册的设备连接管理更多的IP摄像头。
- **注销设备**：在设备不再使用时，注销设备以释放资源。
- **查看设备告警**：您可以通过查看设备告警信息，在端侧处理告警事件。
- **查看设备数据**：在华为HiLens管理控制台左侧导航栏单击“数据管理(Beta)”，进入“数据管理”页面，可选择设备下载并查看此设备相关的所有数据。

6.2 管理设备上的技能

6.2.1 技能管理简介

针对已完成注册且显示在线的设备，您可以在华为HiLens平台管理设备上的技能，包括安装新技能、添加运行时配置、启动技能、停止技能、卸载技能等操作。

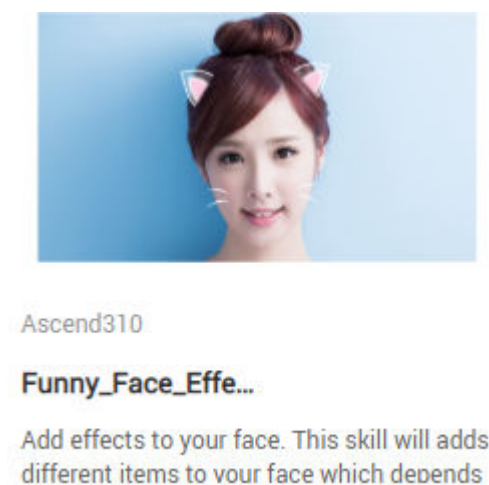
什么是技能

技能（Skill）是运行在端侧摄像头的人工智能应用，一般由模型和逻辑代码组成。其中，逻辑代码是技能的框架，负责控制技能的运行，包括数据读入、模型导入、模型推理、结果输出等；模型是人工智能算法经由大数据训练而成，负责技能运行中关键场景的推理。

华为HiLens的[技能市场](#)提供了丰富的技能供用户选择。

- 按应用场景划分，技能可应用于：智能园区、智慧家庭、智能车载、智能商超和其他等场景。
- 按不同的设备划分，技能分为2种，一种是适用于Ascend芯片的技能，另一种是适用于海思35XX系列芯片的技能。

图 6-1 技能卡片



技能管理说明

- 对技能管理之前，请升级HiLens_Device_Agent固件版本至1.0.7及其以上版本，详细操作请参见[升级HiLens_Device_Agent固件版本](#)。
- 一台设备支持安装多个技能，最多支持安装5个技能。
- 针对新注册的设备，其技能管理页面无任何技能信息，需[安装新技能](#)后使用。
- 由于当前设备管理只支持HiLens Kit设备，设备上的技能管理只能针对“适用于Ascend芯片的技能”。
- 设备上的技能可在技能市场购买、安装等，您可以前往[技能市场](#)获得更多技能。
- 设备上的技能也可自行开发，您可以前往[控制台开发技能](#)。
- 在设备列表单击设备卡片进入设备详情页，默认进入技能管理页签，可查看设备上安装的技能列表，技能状态包含“安装中”、“故障”、“停止”、“运行中”。

支持的操作及其说明

进入“技能管理”界面：在管理控制台左侧菜单栏选择“设备管理 > 设备列表”，然后在设备列表中，选择需要进行管理的某一设备，单击设备卡片进入设备详情页面。“设备详情”页面默认展示“技能管理”页签。

在“技能管理”页签下可以查看安装至当前设备的技能信息，如果技能是容器镜像格式，可单击技能名称左侧的∨查看技能各个实例的状态信息。

- [安装技能](#)
- [添加运行时配置](#)
- [启动或停止技能](#)
- [卸载技能](#)
- **查看设备上技能运行日志：**在“技能管理”页面，单击“更多>查看日志”可跳转至“数据管理”页面，可直接查看此技能运行的实时数据或者运行的历史数据。更多说明可参见[管理数据](#)。
- **为对应技能配置订阅消息：**在“技能管理”页面，单击“更多>技能消息”可跳转至“技能消息”页签，您可以直接为此技能配置订阅对象。更多说明可参见[配置订阅消息](#)。

6.2.2 安装技能

针对您管理的设备，可以在技能市场中选购所需的技能，并将其安装部署到您的设备中，拓展设备的AI能力。

前提条件

- “设备管理>设备列表”中至少拥有一个设备，且设备状态为“在线”。
- “产品订购>订单管理”默认“技能订单”页签和“AI Gallery”页签下中至少拥有一个技能。如果未购买任意技能，即此页面显示无内容，建议您先参考[指导购买技能](#)，在技能市场购买相关技能。

操作步骤

1. 在管理控制台左侧菜单栏选择“设备管理 > 设备列表”，然后在设备列表中，选择需要进行管理的某一设备，单击设备卡片进入设备详情页面。
“设备详情”页面默认展示“技能管理”页签。

图 6-2 技能管理

名称	ID	版本	状态	来源约束	更新时间	到期时间	操作
人脸识别识别_1TB6-385元		1.4.0	停止	1 路	2021/09/25	2021/10/24	启动 停止 卸载 运行时配置 添加实例 更多
安全帽检测_IPC		1.0.7	运行中	1 路	2021/09/22	永久有效	启动 停止 卸载 运行时配置 添加实例 更多

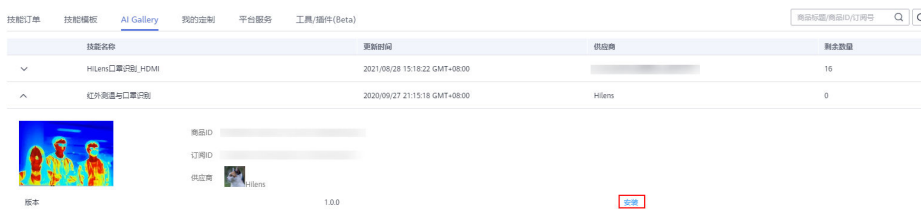
2. 在“技能管理”页签中，单击右上角“添加新技能”，系统跳转至“产品订购>订单管理”页面。
3. 系统默认显示“技能订单”页签，展示您的账号下已完成购买的技能列表。您也可以单击“AI Gallery”，切换至“AI Gallery”页签，展示您账号下在ModelArts AI Gallery购买的技能列表。
4. 选择本次需要安装的技能。
 - “技能订单”页签：单击技能列表“操作”列的“安装”。

图 6-3 技能订单-15



- “AI Gallery” 页签：单击技能名称左侧∨，查看技能信息，单击“安装”。

图 6-4 技能安装



说明

只能选择“适用芯片：Ascend310”类型的技能，且该技能购买的份数未使用完。

5. 在弹出的安装技能对话框中，勾选您需要安装的设备，在“规格”列选择技能在设备上使用的路数，若要安装技能至多台设备，可根据自身需要选择勾选“用此规格批量安装”的“是”，单击“安装”。
6. 执行如下操作确认技能是否正常安装。若技能状态为“安装成功”，说明技能正常安装至设备。
 - a. 单击左侧导航栏“设备管理 >设备列表”，进入设备列表页面。
 - b. 单击已注册设备的“技能管理”，已安装的技能状态为“停止”，单击操作列的“启动”，并单击“确定”，确定启动技能运行在端侧设备上。等待一会，当状态变为“运行中”，则技能成功运行在端侧设备上。

表 6-1 安装技能状态

状态名	状态说明
停止	停止状态，技能停止运行在端侧设备上。
运行中	运行状态，技能成功运行在端侧设备。

6.2.3 添加运行时配置

部分技能在运行的时候，需要用户配置相关参数。比如一般人脸判断的技能都需要用户上传人脸库。

背景信息

- 技能开发者设置了运行时配置的技能，才能够进行相应配置添加和修改。
- “运行时配置”分为两种，“全局”和“视频”配置。
 - “全局”的配置是针对设备整体生效的。

- “视频”是指摄像头分路配置，要求该技能可以处理多路视频。比如人脸判断的技能，每路摄像头的视频都配置一个人脸库，那么该技能就可以处理多路人脸判断。

添加运行时配置

1. 在管理控制台左侧菜单栏选择“设备管理 > 设备列表”，然后在设备列表中，选择需要进行管理的某一设备，单击设备卡片进入设备详情页面。
“设备详情”页面默认展示“技能管理”页签。
2. 在“技能管理”界面中，单击指定的技能卡片。
进入技能详情页。
3. 在“技能管理”页签，单击“操作”列的“运行时配置”。

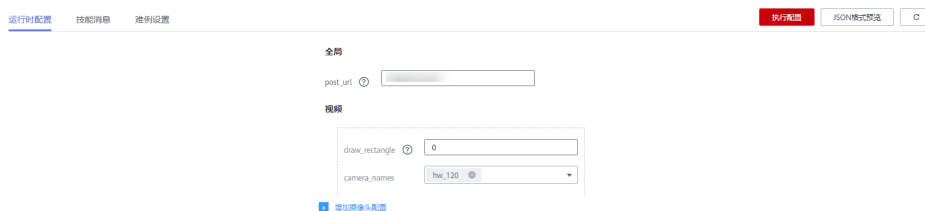
图 6-5 运行时配置



4. 在弹出的对话框中，填写运行时配置的相关参数，相关参数说明可单击参数字段右侧图标查看。

配置参数分为“全局”和“视频”配置。“全局”的配置是针对设备整体生效的。“视频”是指摄像头分路配置，要求该技能可以处理多路视频。比如人脸判断的技能，每路摄像头的视频都配置一个人脸库，那么该技能就可以处理多路人脸判断。

图 6-6 运行时配置-16



6.2.4 配置订阅消息

为指定技能订阅消息后，当技能有输出时，订阅对象会收到邮件或短信通知。

例如某一技能具备检测陌生人并同时发送消息的功能，即该技能在“技能详情”页面的“产品描述”中提示“本技能支持发送订阅消息”，您在安装部署该技能后配置订阅对象，这样当设备检测到陌生人时，所配置的订阅对象将会收到邮件或短信通知。

注意事项

- 首次订阅消息后，需将技能重启，订阅对象才能收到订阅消息，重启操作参见[启动或停止技能](#)。

前提条件

- 技能支持消息通知的功能。由开发者开发技能的时候设定，详细指导和API介绍可参见《开发指南》。

若技能没有配置推送消息功能，普通用户将无法收到技能的消息通知。

- 技能已部署至某一设备中，且处于“运行中”状态。技能非运行状态时，用户无法收到技能的消息通知。

配置订阅对象

订阅对象为消息通知的接收对象，在设备检测到指定场景（例如设备所安装技能具备检测陌生人的功能，在检测到陌生人的场景），订阅对象即可收到消息通知。

1. 登录HiLens管理控制台，单击左侧导航栏“设备管理>设备列表”，进入设备列表页面。
2. 单击技能所部署的设备卡片，默认进入设备详情页的“技能管理”页签。

图 6-7 进入设备详情页



3. 在“技能管理”页签中，单击需要配置订阅对象的“技能名称”，进入技能详情页面。
4. 单击“技能消息”，切换至“技能消息”页签。

图 6-8 技能消息



5. 设置“技能当前主题”。
 - 若添加新的主题
 - i. 单击现有主题右侧+，弹出文本框，在文本框中输入主题名称，可输入字母、数字、下划线或中划线，必须以字母、数字开头，长度不超过255个字符，然后单击✓。
 - ii. 单击主题右侧⊙，在对话框中单击“确定”，确认设置技能主题。
 - 若添加现有主题

单击主题右侧⊙，在对话框中单击“确定”，确认设置技能主题。
6. 单击“技能消息”页签右上侧的“返回技能消息”，配置技能当前主题的信息接收人。

图 6-9 返回技能消息



7. 在订阅列表右侧单击 **+**，弹出“消息接收对象”对话框。

图 6-10 消息接收对象




8. 根据如下情况编辑当前主题的消息接收对象。
 - 若添加新的接收对象
单击“添加消息接收对象”，在弹出的文本框中输入接收人的信息，详细参数解释请参见表6-2。确认信息后单击 **✓**。
 - 若编辑原有接收对象
单击接收对象中的 **✎**，在文本框中重新输入接收人的信息，详细参数解释请参见表6-2。确认信息后单击 **✓**。

表 6-2 消息订阅对象信息

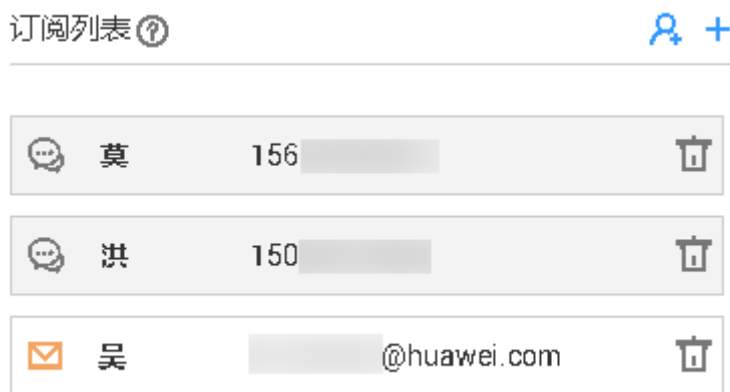
参数名	参数说明
接收人	订阅对象的称呼。
接收方式	接收消息的方式：邮箱或手机。


参数名	参数说明
手机号码/邮箱地址	接收消息的电话或邮箱地址。

可以单击“设为默认”来设置默认消息接收对象，便于快速订阅（在步骤4中的订阅列表右侧单击，即可执行快速订阅）。

- 单击选中订阅该技能的接收人，然后单击“确定”，完成添加消息订阅人。系统跳至订阅列表，可查看刚添加的消息订阅人。若想添加多个消息订阅人，请至步骤7。

图 6-11 订阅列表



- 添加完消息订阅人后，接收人会收到短信或邮件，单击“订阅确认”。反馈订阅成功信息，完成消息订阅。
若订阅对象已确认订阅，您需重启技能，当技能有输出时，订阅对象会收到短信或邮件通知。
同时，在订阅列表中，订阅人图标会由灰色变成橙红色，如图6-11所示。

6.2.5 设置难例

部分技能具备有难例数据收集的功能，在技能运行之前需要做难例参数配置。

前提条件

- 技能开发者设置了难例上传的技能，才能共进行相应的难例设置。
- 在ModelArts创建难例数据集，用于训练难例筛选模型。

设置难例

- 在管理控制台左侧菜单栏选择“设备管理 > 设备列表”，然后在设备列表中，选择需要进行管理的某一设备，单击设备卡片进入设备详情页面。
“设备详情”页面默认展示“技能管理”页签。
- 在“技能管理”界面中，选择指定的技能，单击“操作”列的“运行时配置”。

图 6-12 运行时配置-17

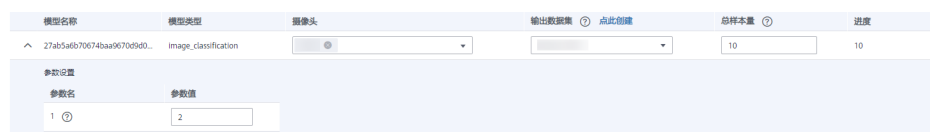


- 单击“难例设置”，在难例设置页签填写相关信息。

表 6-3 难例设置

参数	说明
模型名称	难例筛选模型的名称，用户无需填写。
模型类型	难例筛选的模型类型，比如图像分类、物体检测，用户无需填写。
摄像头	当前模型下选择上传指定摄像头的难例数据。
输出数据集	训练难例筛选模型的数据集。用户需要提前在ModelArts创建难例数据集。
总样本量	输出数据集的样本数量。
进度	当前已经上传的难例数据总量。
参数设置	难例筛选模型需要配置的参数值。如图6-13所示，单击模型名称左侧∨，在“参数设置”下方的“参数值”输入框中填写，单击参数名右侧的ⓘ可查看参数的描述。

图 6-13 难例设置



- 难例设置完成后，单击右上角的“执行配置”。

6.2.6 启动或停止技能

针对设备上的技能，您可以根据您的需求启动或停止技能。启动技能操作成功后，设备上的技能将处于“运行中”状态，表示技能已经在设备中应用。停止操作成功后，设备上的技能将处于“停止”状态。

操作前必读

- 启动或停止技能之前，请升级HiLens_Device_Agent固件版本至1.0.7及其以上版本，详细操作请参见[升级HiLens_Device_Agent固件版本](#)。
- “停止”状态的技能，允许执行“启动”操作。

- “运行中”允许执行停止操作，“故障”或“停止”的技能允许卸载。
- 用户安装技能后，需要启动技能，才能使技能在设备上运行，详情请参见[启动技能](#)。

启动技能

在“技能管理”界面中，选择状态为“停止”的技能，单击“操作”列的“启动”。技能启动时会有一个命令下发过程，您需要等待一段时间才能在界面看到技能状态的更新。

图 6-14 启动技能

名称	ID	版本	状态	资源约束	更新时间	到期时间	操作
人脸识别1-TR6-30天...		1.0.0	停止	1路	2021/09/25	2021/10/25 20:00:00	启动 停止 卸载 运行时配置 添加实例 更多
安全检测1-IPC		1.0.7	运行中	1路	2021/09/22	永久有效	启动 停止 卸载 运行时配置 添加实例 更多

停止技能

在“技能管理”界面中，选择状态为“运行中”的技能，单击“操作”列的“停止”。技能停止后会有一个命令下发过程，您需要等待一段时间才能在界面看到技能状态的更新。

图 6-15 停止技能

名称	ID	版本	状态	资源约束	更新时间	到期时间	操作
人脸识别1-TR6-30天...		1.0.0	运行中	1路	2021/09/25	2021/10/25 20:00:00	启动 停止 卸载 运行时配置 添加实例 更多
安全检测1-IPC		1.0.7	运行中	1路	2021/09/22	永久有效	启动 停止 卸载 运行时配置 添加实例 更多

6.2.7 卸载技能

若不再需要某个技能，可通过HiLens管理控制台卸载技能。

操作前必读

- 仅可对处于“在线”状态（即运行状态）的设备卸载技能。处于离线状态（即非运行状态）的设备无法卸载技能。
- 卸载技能会删除端侧的技能和技能数据，请谨慎操作。

卸载技能

1. 在“技能管理”界面中，进入“技能管理”界面，选择指定的技能，单击“操作”列的“卸载”。

图 6-16 卸载技能

名称	ID	版本	状态	资源约束	更新时间	到期时间	操作
人脸识别1-TR6-30天...	886e480857846548966...	1.0.0	运行中	1路	2021/09/25	2021/10/25 20:00:00	启动 停止 卸载 运行时配置 添加实例 更多
安全检测1-IPC	0062cc9242734862b087...	1.0.7	运行中	1路	2021/09/22	永久有效	启动 停止 卸载 运行时配置 添加实例 更多

2. 在弹出的对话框中，确认要卸载的技能信息，单击“确定”完成卸载操作。

图 6-17 卸载技能-18



6.3 查看设备信息

在设备列表页面，您可以查看已注册设备的基本信息，或者查看此设备升级状态。

查看设备基本信息

在“设备管理>设备列表”页面，默认展示设备的一些基本信息，如图6-18所示。

图 6-18 设备卡片



单击设备对应的卡片的设备名称，进入设备详情页面，如图6-19所示，您可以查看设备的详细情况，并且对设备进行管理，包含技能管理、摄像头管理、升级固件等操作。

图 6-19 设备详情页



6.4 查看设备日志

您可以在使用HiLens技能后，查看设备agent日志和技能日志。

查看方式

- **使用SSH查看日志**：SSH连接至HiLens Kit，查看日志。
- **管理数据**：在华为HiLens管理控制台的“数据管理（Beta）”页面下载并查看设备技能运行的日志。

使用 SSH 查看日志

1. SSH连接到HiLens Kit设备，详细操作指导请参见[连接PC和HiLens Kit](#)。
2. 根据固件版本号，执行查看日志命令：
 - 固件版本1.0.4前（不包含固件版本1.0.4）：
 - 查看设备agent日志
`tail -f /home/hilens/hda/log/hdad.log`
 - 查看技能日志
`tail -f /home/hilens/hda/log/开发者.技能名.技能ID.log`
 - 固件版本1.0.4至1.1.1：
 - 查看设备agent日志
`tail -f /home/log/alog/hilens/hda/hdad.log`
 - 查看技能日志
`tail -f /home/log/alog/hilens/hda/开发者.技能名.技能ID.log`
 - 固件版本1.1.2及以后：
 - 查看设备agent日志
`tail -f /home/log/alog/hilens/hda/hdad.log`
 - 查看技能日志
`tail -f /home/log/alog/hilens/skills/开发者.技能名.技能ID/开发者.技能名.技能ID.log`

说明

为保证您的设备可以运行更新更好的技能，建议及时升级固件版本。升级固件版本请参见[升级HiLens_Device_Agent固件版本](#)。

6.5 升级 HiLens_Device_Agent 固件版本

如果设备生产商推送了新版本的固件，可通过华为HiLens管理控制台升级HiLens Framework版本。随着业务不断演进，工程师将不断升级HiLens Framework版本，为保证您的设备可以运行更新更好的技能，建议及时升级版本。

操作前必读

- 只有“在线”状态的设备才能执行升级固件版本的操作。
- 当“设备列表”中，“固件版本”显示为“可升级”时，表示存在更新版本的固件，您可以选择版本进行升级。
- 在执行升级固件操作时，设备无法执行其他操作。

图 6-20 升级固件版本的要求



升级固件版本

1. 在管理控制台左侧菜单栏选择“设备管理 > 设备列表”。
2. 单击右上角的“批量>升级”，弹出“固件批量升级”对话框。
3. 在对话框中选择设备类型和固件升级的新版本号，下方会显示可升级的设备，包括设备名称、设备类型、固件名称和固件版本。

图 6-21 固件批量升级



4. 勾选需要升级的设备，单击“确定”。
- 在升级过程中，您可以进入设备详情页，查看到固件的版本以及固件升级进度。“固件版本”状态显示“升级成功”，表示升级固件版本成功。

6.6 配置摄像头

HiLens Kit自带一个摄像头，可通过[视频采集器](#)接口打开。同时也可以连接管理多个IP摄像头（摄像头的个数不能大于设备上所安装技能的支持通道数之和），外接摄像头时，需要在控制台添加并配置摄像头。

说明

- HiLens Kit自带摄像头暂时不支持夜视功能。
- HiLens Kit支持接入4K及以下的IPC摄像头，暂不支持接入红外测温摄像头。

添加摄像头

1. 在管理控制台左侧菜单栏选择“设备管理 > 设备列表”，然后在设备列表中单击设备名称，进入设备详情页。
2. 单击“摄像头管理”，切换至设备详情页的“摄像头管理”页签。

图 6-22 摄像头管理



3. 在“摄像头管理”页面中，单击右上角“添加摄像头”，在弹出的对话框中，填写相关信息。

图 6-23 添加摄像头

添加摄像头

* 摄像头名称

用户名

密码

* 协议

* 请求路径

表 6-4 摄像头配置

参数	参数说明
摄像头名称	摄像头的名称，用于标识区分。用户自定义。
用户名	登录IP摄像头时的用户名。从摄像头说明书中获取。
密码	登录IP摄像头的密码。
协议	摄像头传输视频的协议。默认为“rtsp”，且不可修改。
请求路径	访问摄像头视频的URL。例如“192.168.0.1/root”。请从您所外接的摄像头说明书中获取。

4. 确认信息填写完成后，单击“确定”完成摄像头添加，并将配置下发到端侧设备生效。此时摄像头管理页面增加了一行摄像头信息。

编辑摄像头信息

针对已添加到此设备的摄像头，您可以编辑摄像头信息。

1. 在管理控制台左侧菜单栏选择“设备管理 > 设备列表”，然后在设备列表中单击设备名称，进入设备详情页。
2. 单击“摄像头管理”，切换至设备详情页的“摄像头管理”页签。

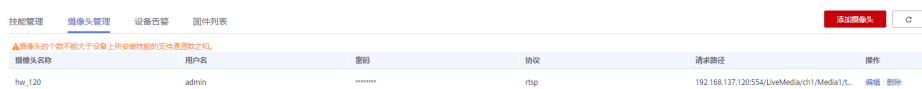
3. 在设备详情页的“摄像头管理”页签中，单击“操作”列的“编辑”，修改摄像头的相关信息，并单击“执行配置”，确定修改摄像头相关信息。
支持修改“用户名”、“密码”、“协议”和“请求路径”信息。“摄像头名称”一旦确定，无法修改。

删除摄像头

针对已添加到此设备的摄像头，您可以删除多余的摄像头。

1. 在管理控制台左侧菜单栏选择“设备管理 > 设备列表”，然后在设备列表中单击设备名称，进入设备详情页。
2. 单击“摄像头管理”，切换至设备详情页的“摄像头管理”页签。

图 6-24 摄像头管理-19



摄像头名称	用户名	密码	协议	请求路径	操作
hw_120	admin	*****	rtsp	192.168.137.120:554/LiveMedia/ch1/Media1_...	编辑 删除

3. 在设备详情页的“摄像头管理”页签中，单击“操作”列的“删除”，然后单击“确定”，确定删除摄像头，即可删除多余的摄像头信息。

说明

删除后，信息不可恢复，请谨慎操作。

6.7 注销设备

注销设备后，华为HiLens云侧与设备解绑，HiLens管理控制台上将没有相应的设备记录，同时失去对设备的控制权。

注销设备后，该设备才能重新注册到新账号下的华为HiLens控制台。

前提条件

如待注销设备已激活，请先解绑激活信息才能注销设备。

在管理控制台左侧菜单栏选择“设备管理 > 设备列表”，单击设备卡片上的“权限激活”，根据页面提示解绑激活信息。

注销设备

1. 在华为HiLens管理控制台，单击左侧导航栏“设备管理 > 设备列表”，进入“设备列表”页面。
2. 选择指定的设备，单击“操作> 注销”。
您也可以单击设备卡片进入设备详情页，单击右上角“注销设备”。

图 6-25 注销设备



3. 在弹出的对话框中，确认要注销的设备信息，单击“确定”完成注销操作。设备注销后，设备上部署的技能将同时被卸载。

📖 说明

注销设备前，需要先卸载从技能市场安装的技能。若不卸载从技能市场上安装的技能就注销设备，设备再次注册时，技能和日志都会被删除。

6.8 查看设备告警

当设备发生故障或因某些原因导致设备处于不正常的工作状态时，华为HiLens能够根据设备不同模块出现的故障产生告警信息，您可以通过查看设备硬件告警信息，在端侧处理告警事件。

华为HiLens也能对业务进行监控，您可以通过[设置告警接收信息](#)，当设备技能无法正常运行时，华为HiLens实时给您发送业务告警信息。

📖 说明

- 设备需要把固件升级到1.0.7或以上版本才支持告警功能，设备需要把固件升级到1.1.0或以上版本才能支持告警通知功能，固件升级请参见[升级HiLens_Device_Agent固件版本](#)。
- 设备只有处于“在线”状态才能将告警信息同步至华为HiLens控制台。

告警级别

- 一般告警（Minor）
一般告警不会对系统产生大的影响，需要尽快采取相应的措施，防止故障升级。
- 严重告警（Major）
严重告警将会对系统产生较大的影响，有可能中断部分系统的正常运行，导致业务中断。
- 紧急告警（Critical）
紧急告警可能会使设备下电，系统中断。需要马上采取相应的措施进行处理。

查看告警步骤

在华为HiLens管理控制台，单击左侧导航栏“设备管理 > 设备列表”，在发生告警事件的设备卡片单击“设备告警”，默认进入“设备告警”页签。

图 6-26 设备告警



您可以在“设备告警”列表查看告警事件的相关信息，并进一步在端侧处理告警。告警信息包括告警的“标题”、“等级”、“内容”、“影响”、“建议”、“产生时间”、“操作”。其中，设备告警可分“一般”、“严重”、“紧急”三个等级，详情请参见[告警级别](#)。

图 6-27 设备告警列表



当您在端侧处理完告警，存在一定时间后，端侧检测到无异常之后，告警会自动清理。

设置告警接收信息

1. 登录HiLens管理控制台，单击左侧导航栏“设备管理>设备列表”，进入设备列表页面。
2. 单击“设备告警”，在“设备告警”页签右侧，单击“告警接收设置”。

图 6-28 告警接收设置



3. 在告警设置下方勾选可实时接收告警信息的告警类别“业务告警”，设置“告警时间段”和“告警频率”。

图 6-29 设备告警-20



- “告警时间段”：设置向订阅对象发送告警的时间段。
- “告警频率”：设置告警频率，在设置的告警时间段内，告警达到指定次数则不再发送消息。例如告警频率设置为“日”、“3”，则每日在设置的告警时间段内如果告警次数达到3次，则不再发送告警信息。

单击业务告警左侧的∨可查看业务告警信息描述和影响。

不支持实时接收“硬件告警”。

4. 在“设备告警”页签右上方单击“保存”。
5. 单击订阅列表右侧+，选择订阅类型（短信/邮件），填写对应的通讯方式（手机号码/邮箱地址），设置告警接收对象。
6. 确认信息后，单击∨。
7. 添加完消息订阅人后，接收人会收到短信或邮件，单击“订阅确认”。
反馈“订阅成功”信息，完成业务告警消息订阅。

删除告警

您可以根据自身情况删除告警信息。

1. 在华为HiLens管理控制台，单击左侧导航栏“设备管理 > 设备列表”，在发生告警事件的设备卡片单击“设备告警”，默认进入“设备告警”页签。
2. 在告警列表中，已处理的告警“操作”列单击“删除”。
弹出“删除告警信息”对话框。

图 6-30 删除告警

序号	标题	等级	内容	影响	建议	产生时间	操作
1	安装在设备上的技能NT...	紧急	多次尝试安装技能均失败	业务中断	查看Log分析原因	2021/09/26 16:10:23 GMT+08:00	删除
2	安装在设备上的技能NT...	紧急	IPC摄像头打开失败	可能影响业务处理	请检查摄像头配置或重启	2021/09/26 14:16:57 GMT+08:00	删除

3. 在对话框中单击“确定”，完成删除告警操作。

图 6-31 删除告警信息

删除告警信息

是否确定删除设备 离线的告警信息？删除该告警后，云侧将不会再提示该告警信息。请确认端侧已经处理完该告警后再删除。



6.9 一键健康检查

由于时区、网络等问题会遇到设备在控制台显示离线、启动技能失败、固件升级失败等问题，您可以通过一键健康检查操作，快速定位并解决问题。

前提条件

云侧一键检查功能仅支持固件为1.1.0或以上版本的设备，升级固件版本请见[升级HiLens_Device_Agent固件版本](#)。

云侧健康检查项

- 进程信息：检查基础进程是否正常。
- 依赖库信息：检查设备依赖库是否被误删、调用是否正常。
- 时间信息：检查时区和时间是否正常。

云侧健康检查步骤

1. 登录华为HiLens管理控制台，在左侧菜单栏选择“设备管理 > 设备列表”。
 2. 在“设备列表”页面单击设备卡片。
进入设备详情页。
 3. 在右上方单击“一键健康检查”。
- 等待一段时间，待系统检查完毕，页面弹出“健康检查”对话框，并显示“进程信息”、“依赖库信息”、“时间信息”的状态。
- 如果状态处于“异常”，“健康检查”对话框会显示异常原因，您可以根据原因进一步解决问题。

图 6-32 一键健康检查



端侧健康检查项

通过SSH登录HiLens Kit设备系统，您也可以针对端侧进行健康检查。

- 检查设备的网络连通性。
- 检查设备连接的无线网络类型是否符合要求。
- 检查设备是否处于离线状态。

端侧健康检查步骤

1. [连接PC和HiLens Kit](#)
2. [SSH登录HiLens Kit设备](#)
3. 执行命令`hdactl health_check`。

7 控制台开发技能

7.1 技能简介

技能可以抽象地理解为算法模型+逻辑代码。算法模型负责关键的AI推理，逻辑代码负责处理模型推理的结果。

本章节介绍在华为HiLens控制台使用空模板和基础技能模板开发技能。

什么是技能

技能（Skill）是运行在端侧摄像头的人工智能应用，一般由模型和逻辑代码组成。其中，逻辑代码是技能的框架，负责控制技能的运行，包括数据读入、模型导入、模型推理、结果输出等；模型是人工智能算法经由大数据训练而成，负责技能运行中关键场景的推理。

华为HiLens的[技能市场](#)提供了丰富的技能供用户选择。

- 按应用场景划分，技能可应用于：智能园区、智慧家庭、智能车载、智能商超和其他等场景。
- 按不同的设备划分，技能分为2种，一种是适用于Ascend芯片的技能，另一种是适用于海思35XX系列芯片的技能。

图 7-1 技能卡片



Ascend310

Funny_Face_Effe...

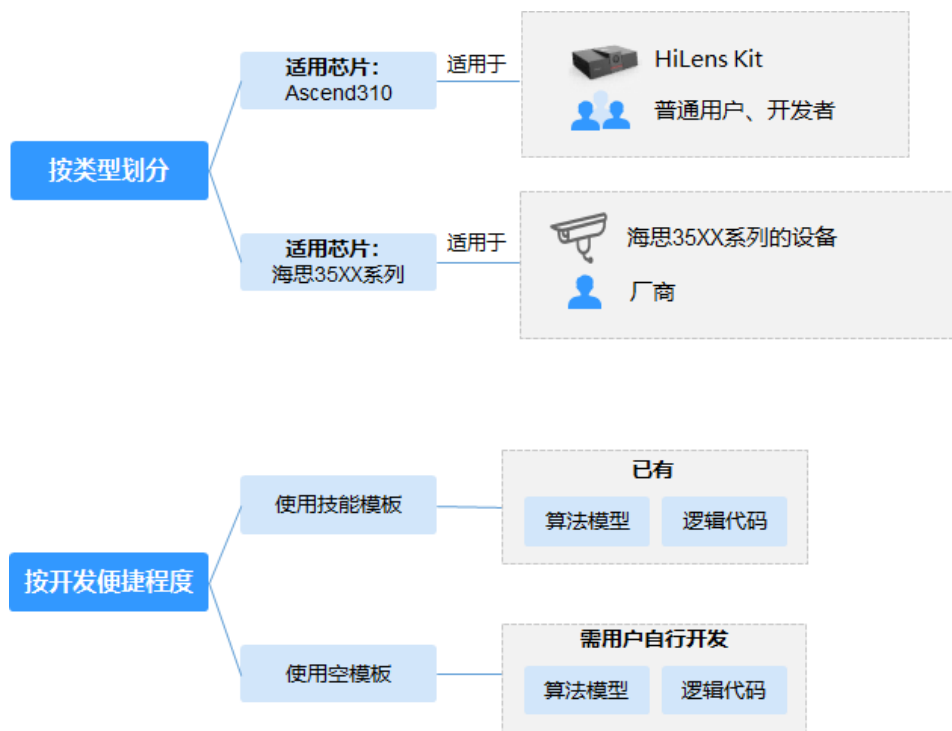
Add effects to your face. This skill will adds different items to your face which depends

技能开发说明

根据不同芯片类型，技能分为两种类型，分别为可应用于HiLens Kit设备的技能和可应用于海思35XX系列芯片设备的技能。平台不仅支持使用技能模板快速开发技能，便捷高效；还支持自行开发模型和逻辑代码，可以满足您更丰富的场景。

运行在海思35XX系列芯片的技能对内存和性能的要求比较高，针对适用于此类设备的技能需要对模型进行优化后才能运行，若需要开发此类技能，请联系华为工程师。

图 7-2 开发技能



控制台新建技能的流程

在准备开始新建技能前，请明确您需要开发一个什么类型的技能，不同类型的技能，其主要区别在于模型和逻辑代码，与新建技能的流程无关。

说明

在华为HiLens控制台开发技能时，会占用OBS资源，需要收取一定费用，收费规则请参见[对象存储服务OBS](#)。

新建技能的详细流程如[图7-2](#)所示。

- **使用基础技能模板：**需确保平台已有可用的基础技能模板。
- **使用空模板：**需开发您的技能所需的模型以及逻辑代码，如果您的模型格式不符合平台要求，华为HiLens平台会针对不符合要求的模型，在导入（转换）模型操作中进行模型转换。

图 7-3 技能开发

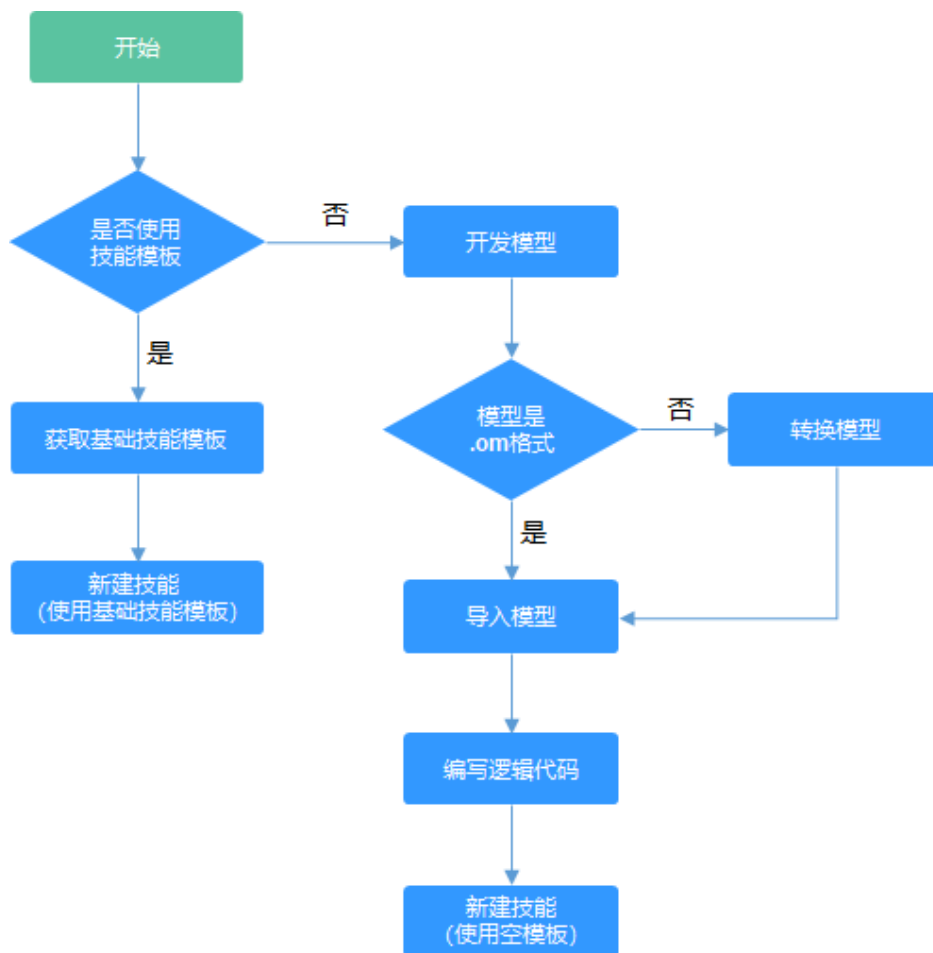


表 7-1 新建技能常用操作指引

类别	常用操作	快速跳转至详细指导
使用技能模板	获取技能模板	获取技能模板
	新建技能（使用基础技能模板）	新建技能
使用空模板	开发模型	开发算法模型
	导入（转换）模型	导入（转换）模型
	编写逻辑代码	编写逻辑代码
	新建技能（使用空模板）	新建技能

技能相关操作

在华为HiLens平台，在开发技能的同时，您可以管理您开发的技能，包含的操作如下表所示。

表 7-2 技能相关操作

操作	说明
新建技能	在确定您需要开发的技能类型后，根据 控制台新建技能的流程 新建一个您所需的技能。
编辑技能	技能新建完成后，您可以编辑修改技能，包括基本信息、技能内容和运行时设置。
安装和调试技能	在技能新建完成后，您可以将技能部署至设备中，通过查看技能视频确定技能效果，反复调试技能。
删除技能	对于不再使用的技能，您可以删除技能以释放资源。

7.2 新建技能

7.2.1 使用技能模板

本章节介绍使用基础技能模板新建技能的操作步骤。使用基础技能模板新建技能，可以使用基础技能模板中的默认值，包含逻辑代码或算法模型等，只需填写基本信息即可完成一个技能的创建。

前提条件

- 保证华为云账号不欠费。在华为HiLens控制台开发技能时，会占用OBS资源，需要收取一定费用，收费规则请参见[对象存储服务OBS](#)。
- 已在技能模板页面查找并获取到满足相应业务要求的技能模板，详细操作请参见[获取技能模板](#)。

1. 开始新建技能

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>技能管理”，进入技能列表。
2. 在“技能管理”页面，单击右上角“新建技能”，进入“创建技能”页面。

图 7-4 创建技能



2. 选择模板

1. 在“创建技能”页面，在“技能模板”中单击“选择已有模板”，弹出“选择技能模板”对话框。

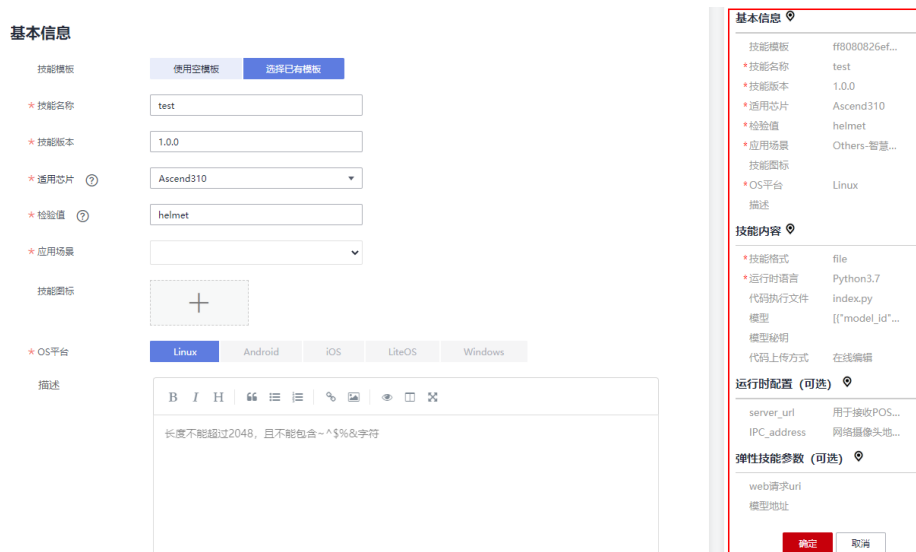
图 7-5 选择技能模板

选择技能模板



2. 在弹出的对话框中，选择您想要使用的模板，然后单击“确定”。
此时，页面自动跳转至“创建技能”页面的“技能内容”编辑区域，您选择的模板的默认配置将自动加载，您可以在创建技能页面右侧查看到相关信息。

图 7-6 技能基本信息



3. 填写相关信息

选择已有模板后，您可以参考[使用空模板](#)章节的指导，填写技能相关信息，包含[基本信息](#)、[技能内容](#)、[运行时配置（可选）](#)。

其中，“检验值”会根据所选模板自动初始化，“技能名称”、“技能版本”等参数是您必须填写的，技能模板并未携带此类信息。针对技能模板中的内容，您可以根据自身业务情况进行修改，也可以保持其默认值。

4. 确认信息并完成新建技能

上述参数填写完成后，您可以在界面右侧查看其配置参数值，如果某个字段填写错误，在右侧会显示一个小红叉。

确认信息无误后，单击“确定”完成技能创建。

创建完成后，您的技能将进入“技能开发>技能管理”页面。

发布后，技能将处于“审核中”状态，华为HiLens工作人员将在3个工作日内完成审核，请您耐心等待。当审核通过后，状态将变更为“已发布”。

7.2.2 使用空模板

本章节介绍使用空模板新建技能的操作步骤。如果您选择使用空模板，需提前准备好使用的模型和逻辑代码。

背景信息

- 在华为HiLens控制台开发技能时，会占用OBS资源，需要收取一定费用，收费规则请参见[对象存储服务OBS](#)。
- 算法模型必须是.om格式，且满足华为HiLens的要求，详细可参考[开发算法模型](#)。
- 逻辑代码的存储方式有两种：“在线编辑代码”和“从OBS上传代码”。
 - 如果您的代码逻辑较简单，建议直接在线编辑代码。

- 如果您的代码结构比较复杂，推荐用本地IDE开发完成后，使用OBS上传代码zip或tar.gz包的方式，上传操作可参见[OBS快速入门](#)。上传到OBS需要收取一定费用，收费规则请参见[对象存储服务OBS](#)。

📖 说明

- 上传文件时，要求此OBS桶与您使用的华为HiLens处于同一区域。
- 若您从OBS上传代码，需要将所有代码文件压缩后上传，上传的文件必须为“.zip”或“.tar.gz”格式，且入口文件必须在一级目录。如下示例所示，入口代码（“main.py”）在一级目录，其它的代码按需求自行设计，您可将模型与代码一起打包上传。
- 入口代码所在文件（如上面的main.py）通过参数【代码执行文件】进行配置，后面会有说明。

代码目录示例如下所示：

```
skill/
|---main.py      #入口文件，名称与新建技能的“代码配置”参数一致
|---depends/     #可选，用于存放模型文件
|---workspace/  #工作空间，用于存放技能生成的数据
|---data/       #用于存放运行时配置
```

- 若技能格式为容器镜像格式，请先将镜像上传至容器镜像服务（SWR），上传方式请参见[Docker客户端上传镜像](#)。

前提条件

- 保证华为云账号处于不欠费状态。在华为HiLens控制台开发技能时，会占用OBS资源，需要收取一定费用，收费规则请参见[对象存储服务OBS](#)。
- 新建技能使用的算法模型已[导入HiLens](#)中。
- 如果使用“从OBS上传代码”方式，请提前按照要求将逻辑代码上传至OBS桶中。

1. 开始新建技能

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>技能管理”，进入技能列表。
2. 在“技能管理”页面，单击右上角“新建技能”，进入“创建技能”页面。

图 7-7 创建技能-21



2. 基本信息

在“创建技能”页面，在“技能模板”中选择“使用空模板”后，填写基本信息。参数说明请参见[表7-3](#)。

图 7-8 填写基本信息

基本信息

技能模板 使用空模板 选择已有模板

* 技能名称

* 技能版本

* 适用芯片

* 检验值

* 应用场景

技能图标

* OS平台 Linux Android iOS LiteOS Windows

描述

B I H | “ ” | ≡ | ≡ | 🔗 | 🖼️ | 👁️ | 🗑️ | ✕

长度不能超过2048，且不能包含~^\$%&字符

表 7-3 基本信息参数设置

字段名称	说明
技能模板	是否使用模板来开发技能。如果选择已有模板，则会使用模板的模型和代码。即技能内容模块的字段都会使用模板的信息。
技能名称	技能的名称，下发到端侧也使用该名称作为根目录。 可输入字母、数字、下划线或中划线，必须以字母开头，以字母或数字结尾，长度3~60个字符。 说明 由于技能市场不允许存在同名技能，所以如果想发布到技能市场，请使用全局唯一的名称对您的技能进行命名。
技能版本	技能的版本。版本号格式为“A.A.A”，“A”为不超过3位的自然数。例如“1.0.0”。 当“A”超过一位时，不允许以“0”开头，比如禁止输入“01.0.0”。
适用芯片	技能支持的芯片。运行在HiLens Kit上的技能需选择Ascend 310芯片。

字段名称	说明
检验值	用于技能校验，防止技能伪装。要与代码中初始化接口的参数一致。具体使用方法请参见开发指南>初始化HiLens Framework。
应用场景	技能适用的场景，例如“智慧园区”、“智慧家庭”、“智能车载”、“智能商超”的子场景及“其他”。
技能图标	技能的头像。
OS平台	技能运行的操作系统平台，包括Linux、Android、IOS、LiteOS和Windows。HiLens Kit用的是基于CentOS制作的Linux发行版Euler系统操作系统，所以如果开发的是运行在HiLens Kit的技能，这里选择Linux。
描述	对技能的详细介绍。 最大长度2048，不允许输入~^\$%&等特殊字符。 说明 若新建技能的场景需要发送消息到用户的手机或邮箱，比如检测到陌生人技能需要在发现陌生人后发送消息提醒用户，描述中 必须注明 ：“本技能支持发送订阅消息”，发送的具体内容需在 编写逻辑代码 时注明，详细指导和API介绍可参见开发指南。

3. 技能内容

根据您的模型和逻辑代码情况，填写技能内容，详细参数说明请参见表7-4。

图 7-9 技能内容

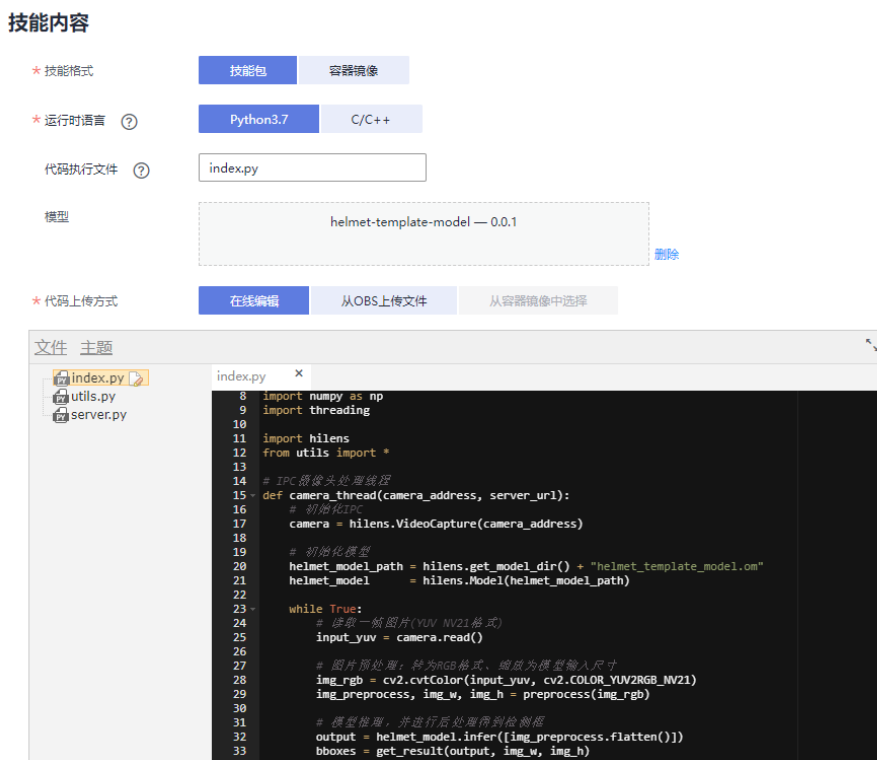


表 7-4 技能内容的参数说明

字段名称	说明
技能格式	技能的格式。可分为“技能包”和“容器镜像”两种格式。
运行时语言	逻辑代码的运行时语言。目前支持“Python3.7”和“C/C++”。如果选择C/C++语言开发，将不支持在线编辑代码。开发者线下开发完成后，需要在Linux环境下把代码编译打包，然后上传到obs。
代码执行文件	技能的启动将以该代码执行文件为入口，类似C语言的main函数。入口文件必须在一级目录。 <ul style="list-style-type: none"> “技能包”格式的技能要求该文件在技能包的根目录下，如Python技能请以.py结尾。 “容器镜像”格式的技能，如果dockerfile有配置ENTTRYPOINT，概述如可以为空，否则要求输入该镜像内执行文件的路径。 文件名不超过1024位，允许输入字母、数字或下划线。不允许输入#~^等字符。
模型	技能的核心算法。在本地或ModelArts训练， 导入到HiLens平台 。单击加号，您可以在弹出框中，选择华为HiLens平台模型管理列表下的模型。
代码上传方式	逻辑代码的上传方式。 说明 固件1.1.2及以后版本支持配置技能的Python依赖库。开发者在开发技能过程中，可根据自身需要配置技能的Python依赖库，详情请见 配置技能的Python依赖库 。 <ul style="list-style-type: none"> “在线编辑” 使用在线开发技能的逻辑代码，会自动生成一个zip包上传到OBS中。HiLens会自动创建一个OBS桶用于存储技能包，桶名称命名规则为“<project_id>-hilens-skill”，您可以前往OBS服务，找到对应命名规则的OBS桶，并获取在线编辑的代码包文件。 在线编辑时，可在编辑区创建文件或文件夹，并在文件中编辑逻辑代码。默认创建初始入口文件“index.py”，可重命名文件夹或文件名称，也可根据自身业务创建非“.py”格式的文件或文件夹。 “从OBS桶上传文件” 针对比较复杂的工程代码可以线下开发，手动生成zip或tar.gz包并上传到OBS中，然后在此参数中选择使用OBS中的zip或tar.gz代码包。在上传OBS之前，您可以把模型一起打包并上传使用，模型与代码结构可参考背景信息中定义的规范。 “从容器镜像中选择” 针对容器镜像格式的技能，可以提前将容器镜像文件上传至SWR，方式请参见Docker客户端上传镜像。

4. 运行时配置（可选）

图 7-10 运行时配置-22



部分技能在运行的时候需要用户配置参数，比如人脸判断类的技能需要用户上传人脸库等。运行时配置就像是一个“钩子”，开发者把“钩子”放出去，用户运行技能的时候设置了这些配置项，HiLens就会帮开发者把“钩子”收回来，这时候在代码里面开发者就可以使用这些用户的设置。您可以在界面中单击添加配置标签，参考表7-5填写您的配置。

您还可以单击“预览JSON格式”查看“钩子”的格式。开发者通过HiLens Framework提供的get_skill_config接口获取技能配置的JSON格式，读取里面字段的值来使用用户的配置。

运行时配置JSON格式如下所示：

```
{
  //全局配置
  "Language": "enum",
  //视频分路配置
  "multi_camera": [ //该字段key值multi_camera固定，值表示摄像头分路配置
    {
      "camera_names": [ //该字段key值camera_names固定，值表示该分组配置里包含了哪些摄像头
        "cameraX",
        "cameraY"
      ],
      "FaceLib": { //配置名
        "from": "file source", //文件类型值，表示文件来自哪个服务。比如OBS
        "path": "file path" //文件类型值，表示文件的路径
      }
    }
  ],
}
```

表 7-5 运行时配置的参数说明

字段名称	说明
配置名	配置的名称，逻辑代码里面可用该字段获取该配置的值。运行时配置指技能在运行的时候，由用户配置的内容。比如人脸判断技能需要用户上传人脸库等。需要说明的是，整个运行时配置是可选的，但如果添加了一条配置，那么“配置名”、“配置说明”是必填项。
值类型	表示该配置项的值是什么类型的数据。 数值类型（int、float）、枚举类型、字符串和文件类型。

字段名称	说明
值约束	<p>表示该配置可以设置的范围。跟值类型有关。</p> <ul style="list-style-type: none"> 数值类型：用开闭区间的表示方法。如[1,100)表示大于等于1小于100。 字符串和文件：该项无需输入。 枚举：集合的表示方式，如{a,b}。
配置说明	<p>说明该配置的作用，以及配置方法。该字段非常重要，用户会根据该字段的说明来配置，所以如果有运行时配置，开发者一定要在这里解释清楚。</p>
配置对象	<p>HiLens Kit可以连接多个（个数不大于设备上所安装技能的支持通道数之和）摄像头，所以可以开发支持多路视频的技能。</p> <ul style="list-style-type: none"> 选择“视频”，则表示该配置是可以针对设备上每个摄像头视频进行配置的，比如摄像头1和摄像头2采用不同的人脸库。 选择“全局”，则表示设备上所有摄像头都用同一个配置值。

6. 确认信息并完成新建技能

上述参数填写完成后，您可以在界面右侧查看其配置参数值，如果某个字段填写错误，在右侧会显示一个小红叉。

确认信息无误后，单击“确定”完成技能创建。

图 7-11 确认信息并完成空模板新建技能

基本信息 📍

技能模板	使用空模板	
* 技能名称		✖
* 技能版本		✖
* 适用芯片	Ascend310	
* 检验值	hello	
* 应用场景	智能园区-行...	
技能图标		
* OS平台	Linux	
描述		

技能内容 📍

* 技能格式	file	
* 运行时语言	Python3.7	
代码执行文件	index.py	
模型	[]	
模型密钥		
代码上传方式	在线编辑	

运行时配置 (可选) 📍

弹性技能参数 (可选) 📍

web请求uri	
模型地址	
请求方法	POST
计算资源	GPU

确定

取消

创建完成后，您的技能将进入“技能开发>技能管理”页面。

7.3 获取技能模板

技能模板可用于快速新建技能，提升开发效率。

本章节介绍如何获取和使用技能模板。

技能模板介绍

基于基础模板，还可以执行如下操作：

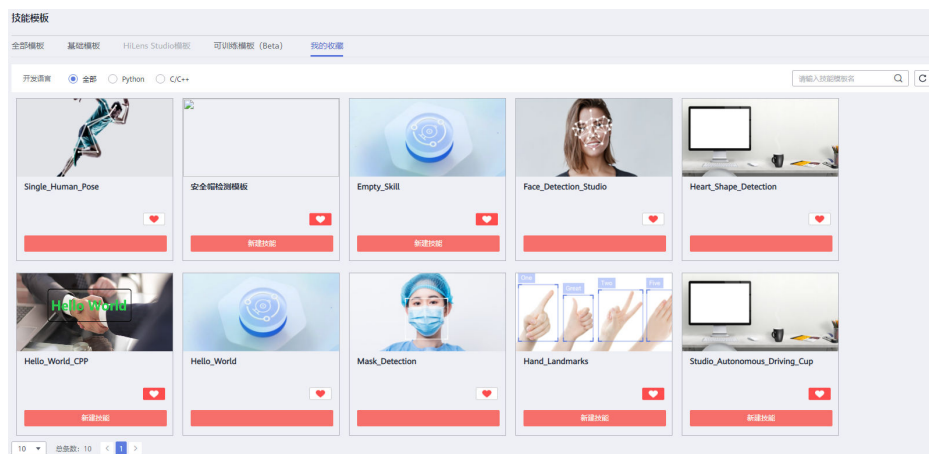
- [查看技能模板](#)
- [下载技能模板](#)
- [收藏技能模板](#)
- [使用技能模板新建技能](#)

查看技能模板

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>技能模板”，默认进入“全部模板”页面。
2. 在“全部模板”页面中，单击技能模板的卡片，可进入技能详情页面，您可以查看技能的基本信息、运行时参数等信息。

您可以单击“我的收藏”页签，查看您收藏的技能列表，也可在此页面单击技能卡片进入详情页。

图 7-12 技能模板



下载技能模板

1. 在“全部模板”、“基础模板”或“我的收藏”页面中，单击技能模板的卡片，可进入技能详情页面。
2. 在技能详情页面，单击右上角的“下载模板”，技能模板将以压缩包的形式下载至您的本地。

压缩包为“.zip”格式。下载的压缩包中包含了此技能模板的算法模型和逻辑代码，有些技能模板可能不存在算法模型，此时压缩包中仅有逻辑代码文件。


图 7-13 下载技能模板




收藏技能模板

图 7-14 收藏技能模板



- 在“全部模板”页面中，单击技能模板的卡片，进入技能详情页面。在技能详情页面，单击右上角的“收藏模板”。
- 在“全部模板”页面中，单击技能模板的卡片的  图标即可收藏模板。

技能模板收藏后，将显示在“我的收藏”页面中。再次单击卡片上的  图标，或技能详情右上角的“取消收藏”按钮，即可取消收藏该模板。

使用技能模板新建技能

1. 在“全部模板”、“基础模板”或“我的收藏”页面中，单击技能模板的卡片，可进入技能详情页面。
2. 在技能详情页面，单击右上角的“用模板新建技能”跳转至“新建技能”页面。新建技能的详细的参数填写请参见[使用技能模板](#)。

7.4 管理算法模型

7.4.1 开发算法模型

前提条件

目前，HiLens Kit的AI芯片支持运行“.om”模型，“.om”模型可以通过TensorFlow或Caffe模型转换而来，但“.om”模型并不支持TensorFlow和Caffe全部的算子，所以在开发模型的时候开发者需要用“.om”模型支持的算子，才能把TensorFlow和Caffe

模型转换成“.om”模型。“.om”模型支持的TensorFlow和Caffe算子边界请见附录[Caffe算子边界](#)和[Tensorflow算子边界](#)。

📖 说明

“.om”模型当前暂不能完全兼容TensorFlow内置的Keras API。

“.om”模型当前不支持Caffe2。

采用 ModelArts 开发

ModelArts是面向AI开发者的一站式开发平台，您可以使用ModelArts开发一个用于HiLens平台的算法模型。

在使用ModelArts开发或训练的模型后，其模型将存储在ModelArts的模型列表中。由于目前华为HiLens仅支持TensorFlow和Caffe引擎开发的算法模型，因此建议在使用ModelArts开发时，注意选择引擎类型以及模型保存格式。

ModelArts自动学习功能训练生成的模型，暂时不支持用于HiLens平台。

线下开发

线下开发指您在本地使用自己熟悉的算法模型开发工具，开发算法模型。

当前仅支持TensorFlow和Caffe引擎开发的算法模型，且您开发的模型需保存为“.pb”或“.caffemodel”格式。然后再使用导入（转换）模型功能将其转换为Ascend310芯片支持的“.om”格式。

后续操作

模型开发完成后，需要把模型导入至华为HiLens控制台。针对TensorFlow和Caffe引擎开发的算法模型，会通过导入（转换）模型功能将其转换为Ascend310芯片支持的“.om”格式。具体操作请参见[导入（转换）模型](#)。

7.4.2 导入（转换）模型

技能可以抽象地理解为算法模型+逻辑代码。算法模型负责关键的AI推理，逻辑代码负责处理模型推理的结果。因此在开发技能前，需要将模型导入华为HiLens。

模型要求

导入的模型可为“.om”格式、“.pb”格式或“.caffemodel”格式，其中“.om”格式的模型才可以在HiLens Kit上运行，“.pb”格式或“.caffemodel”格式的模型在“导入（转换）”过程中，华为HiLens平台会自动将模型转换为“.om”格式。

并非所有模型都能转换成功，进行导入（转换）模型操作前，请确认是否为“.om”模型支持的TensorFlow和Caffe算子边界，详情请见附录[Caffe算子边界](#)和[Tensorflow算子边界](#)。

前提条件

在导入模型前，导入的模型可通过ModelArts在线训练，也可通过本地训练。

- 本地训练模型

导入自定义模型前，需要将自定义的模型上传到OBS服务，非“.om”格式的模型上传文件包含caffe模型文件“.caffemodel”和“.prototxt”和配置文件“.cfg”，或tensorflow的“.pb”模型文件和配置文件“.cfg”。

上传操作可参见[OBS快速入门](#)。模型文件上传至OBS的目录需满足一定规范，详情请见[模型输入目录规范](#)。

上传到OBS需要收取一定费用，收费规则请参见[对象存储服务OBS](#)。

上传至OBS时，需确保OBS桶与华为HiLens在同一区域，且OBS文件夹名称满足如下要求：

- 文件夹名称不能包含以下字符：\:*?"<>|。
- 文件夹名称不能以英文句号(.)或斜杠(/)开头或结尾。
- 文件夹绝对路径总长度不超过1023个字符。
- 文件夹名称不能包含两个以上相邻斜杠(/)。

导入（转换）模型操作

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>模型管理”，进入“模型管理”页面。
2. 在“模型管理”页面，单击右上角的“导入（转换）模型”。
3. 在“导入模型”页面，然后参考[表7-6](#)填写参数，信息确认无误后单击“确定”完成导入。

图 7-15 导入模型

表 7-6 导入自定义模型

参数名称	参数说明
名称	导入模型的名称。模型名称仅支持输入小写字母、数字、中划线字符，输入以小写字母开头，以小写字母、数字结尾，长度2~24的字符。
版本	导入模型的版本号。 格式为“数字.数字.数字”，如2.1.3，每位数字范围为0~999，超过1位时，不能以0开头。

参数名称	参数说明
描述	导入模型的描述。仅支持输入长度1~100的字符。不允许输入 &! ""\<>=等特殊字符。
模型来源	<p>选择导入模型的来源，包括“从OBS导入”和“从ModelArts导入”。</p> <ul style="list-style-type: none"> 从OBS导入 单击“从OBS导入”，选择导入自定义模型存储在OBS服务的桶和文件夹，单击“确定”。 从ModelArts导入 <ol style="list-style-type: none"> 单击“从ModelArts导入”，在右侧下拉框中选择模型框架，包括“TensorFlow”、“Caffe”、“OM（从转换任务中获取）”。 在您进行导入（转换）操作后，华为HiLens平台将TensorFlow、Caffe模型转换为“.om”格式，然后导入模型。OM模型是从前者转换任务中获取。 在下方的模型列表中勾选待导入的模型。
转换类型	<p>若在上文“模型来源”中，选择需要转换格式的模型（非“.om”格式模型），需要选择模型转换的类型，包括“TF-FrozenGraph-To-Ascend-HiLens”、“TF-SavedModel-To-Ascend-HiLens”、“TF-FrozenGraph-To-Ascend”、“TF-SavedModel-To-Ascend”、“Caffe转Ascend”。</p> <ul style="list-style-type: none"> TF-FrozenGraph-To-Ascend-HiLens 该模板将TF frozen_graph模型转换成可在ascend芯片上运行的模型。HiLens Kit系统固件版本为2.2.200.011时，建议使用此模板进行转换。 TF-SavedModel-To-Ascend-HiLens 该模板将TF saved_model模型转换成可在ascend芯片上运行的模型。HiLens Kit系统固件版本为2.2.200.011时，建议使用此模板进行转换。 TF-FrozenGraph-To-Ascend 该模板将TF frozen_graph模型转换成可在ascend芯片上运行的模型。HiLens Kit系统固件版本低于2.2.200.011时，建议使用此模板进行转换。 TF-SavedModel-To-Ascend 该模板将TF saved_model模型转换成可在ascend芯片上运行的模型。HiLens Kit系统固件版本低于2.2.200.011时，建议使用此模板进行转换。 Caffe转Ascend 支持将Caffe模型转换成可在ascend芯片上运行的模型。 TF-FrozenGraph-To-Ascend-893 该模板将TF frozen_graph模型转换成可在ascend芯片上运行的模型。HiLens Kit系统固件版本低于2.2.200.011时，建议使用此模板进行转换。

参数名称	参数说明
转换输入目录	若在上文“模型来源”中，选择需要转换格式的模型（非om格式模型），转换输入目录根据模型的来源自动初始化。目录下应该包含模型文件和转换所需的配置文件。模型文件的输入目录规范请见 模型输入目录规范 。
转换输出目录	若在上文“模型来源”中，选择需要转换格式的模型（非om格式模型），单击 <input type="checkbox"/> 选择模型转换完成后输出的OBS目录。请确保该目录下没有om模型。输出目录规范请见 模型输出目录规范 。
高级选项	包括输入张量形状、转换输出节点等参数选项，详情请见 表 7-7 。

表 7-7 高级选项

参数名称	参数说明
输入张量形状	<p>若在上文“模型来源”中，选择需要转换格式的模型（非om格式模型），并且转换类型是“Tensorflow frozen graph 转 Ascend”或“Tensorflow SavedModel 转 Ascend”时，需要填写输入张量形状。</p> <p>张量形状即模型输入数据的shape，输入数据格式为NHWC，如“input_name:1,224,224,3”。</p> <p>“input_name”必须是转换前的网络模型中的节点名称。当模型存在动态shape输入时必须提供。例如“input_name1:?,h,w,c”，该参数必填，其中“?”为batch数，表示1次处理的图片数量，需要根据实际情况填写，用于将动态shape的原始模型转换为固定shape的离线模型。</p> <p>如果存在多个输入，请以分号(;)隔开。</p>
转换输出节点	<p>指定输出节点,例如 “node_name1:0;node_name1:1;node_name2:0”，其中“node_name”必须是模型转换前的网络模型中的节点名称,冒号后的数字表示第几个输出,例如 “node_name1:0”，表示节点名称为 “node_name1”的第0个输出。</p>
输入数据格式	默认是“NHWC”，如果实际是“NCHW”的话,需要通过此参数指定“NCHW”。
优选数据格式	<p>指定网络算子优先选用的数据格式，“ND(N=4)”和“5D”。仅在网络中算子的输入数据同时支持“ND”和“5D”两种格式时，指定该参数才生效。“ND”表示模型中算子按“NCHW”转换成通用格式，“5D”表示模型中算子按华为自研的5维转换成华为格式。“5D”为默认值。</p>

参数名称	参数说明
生成高精度模型	指定是否生成高精度“FP16 Davinci”模型。 <ul style="list-style-type: none"> 0为默认值，表示生成普通“FP16 Davinci”模型，推理性能更好。 1表示生成高精度“FP16 Davinci”模型，推理精度更好。
网络输出数据类型	“FP32”为默认值,推荐分类网络、检测网络使用;图像超分辨率网络,推荐使用“UINT8”,推理性能更好。

模型导入后，将进入“模型管理”页面，您导入的模型可从列表中查看。

查看模型详情

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>模型管理”，进入“模型管理”页面。

您可以在列表中查看“模型名称”、“版本”、“模型大小”、“导入时间”、“状态”、“描述”和“操作”。其中，模型状态包括“转换中”、“转换失败”、“转换成功”、“导入失败”、“导入成功”。

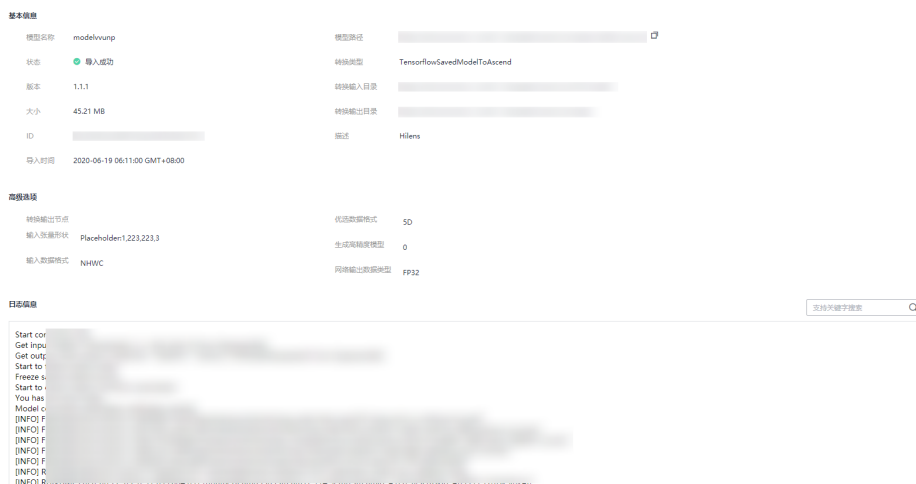
2. 单击操作列的“查看详情”，进入“模型详情”页面。

您可以查看模型的“基本信息”和“日志信息”，如图7-16所示。

“基本信息”的参数说明请见表7-7。

针对需要转换的模型，您可以在“日志信息”区域右上方通过关键字搜索快速定位日志中关键信息的位置。

图 7-16 模型详情



重新转换模型

当您导入的模型格式不是“.om”格式，且模型状态处于“转换失败”的状态时，您可以修改模型的相关参数，重新转换模型。

模型状态可在华为HiLens管理控制台的“模型管理”页面查看。

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>模型管理”，进入“模型管理”页面。
2. 单击操作列的“查看详情”，进入“模型详情”页面。
您可以查看模型的“基本信息”和“日志信息”。
针对重新转换的模型，您可以在“日志信息”区域右上方通过关键字搜索快速定位日志中关键信息的位置。
3. 单击对“基本信息”中的“转换类型”、“高级选项”、“转换输入目录”、“转换输出目录”、“描述”进行修改编辑，编辑完成后单击保存信息。参数说明请见表7-7。
4. 修改完模型参数后，单击右上角“重新转换”，即可重新转换模型。

删除模型

您可以根据自身业务需要，删除已导入（转换）的模型。状态为“转换中”的模型不可执行删除操作。

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>模型管理”，进入“模型管理”页面。
2. 单击操作列的“删除”，完成删除模型操作。也可单击操作列的“查看详情”，在“模型详情”页面的右上角单击“删除模型”，完成删除模型操作。

图 7-17 删除模型

模型名称	模型大小	导入时间	备注	状态	描述	操作
model0445a272-0	31.60 MB	2021/04/17 14:59:56 GMT+08:00		导入成功	import model	查看详情 删除
model0895a95f-1	31.60 MB	2021/04/17 09:54:31 GMT+08:00		导入成功	import model	查看详情 删除
model04810e1191-1	31.60 MB	2021/04/16 17:59:59 GMT+08:00		导入成功	import model	查看详情 删除

7.4.3 模型输入目录规范

在本地开发模型后或在ModelArts开发并转换模型后，针对Ascend芯片，其模型输入目录需要满足一定的规范要求。华为HiLens当前对模型输入目录的要求如下：

- 针对基于Caffe框架的模型，执行模型导入（转换）时，其输入目录需符合如下规范。

```
|
|---xxxx.caffemodel      模型参数文件，输入目录下有且只能有一个，必填。
|---xxxx.prototxt       模型网络文件，输入目录下有且只能有一个，必填。
|---insert_op_conf.cfg   插入算子配置文件，输入目录下有且只有一个，可选。
|---plugin               自定义算子目录，输入目录下有且只能有一个plugin文件夹，可选。仅支持基于TE
                        (Tensor Engine) 开发的自定义算子。
```

- 针对基于TensorFlow框架的模型（“frozen_graph”或“saved_model”格式），执行模型导入（转换）时，其输入目录需符合如下规范。

“frozen_graph”格式

```
|
|---xxxx.pb              模型网络文件，输入目录下有且只能有一个，必填。支持以frozen_graph或
                        saved_model格式保存的模型。
|---insert_op_conf.cfg   插入算子配置文件，输入目录下有且只有一个，可选。
|---plugin               自定义算子目录，输入目录下有且只能有一个plugin文件夹，可选。仅支持基于TE
                        (Tensor Engine) 开发的自定义算子。
```

“saved_model”格式

```
|
|---saved_model.pb       模型网络文件，输入目录下有且只能有一个，必填。支持以frozen_graph或
                        saved_model格式保存的模型。
```

```

|---variables      固定子目录名称，包含模型的权重偏差等信息，必选
|   |---variables.index 必选
|   |---variables.data-00000-of-00001 必选
|---insert_op_conf.cfg  插入算子配置文件，输入目录下有且只有一个，可选。
|---plugin          自定义算子目录，输入目录下有且只能有一个plugin文件夹，可选。仅支持基于TE
                    (Tensor Engine) 开发的自定义算子。
    
```

7.4.4 模型输出目录规范

模型导入（转换）任务执行完成后，华为HiLens将转换后的模型输出至指定的OBS路径。针对不同的转换任务，基于Ascend芯片，其模型输出目录需要满足一定的规范要求。华为HiLens当前对模型输出目录的要求如下：

- 针对基于Caffe框架的模型，执行模型导入（转换）时，其输出目录说明如下所示。

```

|
|---xxxx.om      转换输出的模型，可用于Ascend芯片，模型文件后缀统一为“.om”。
|---job_log.txt  转换过程的日志文件。
    
```

- 针对基于TensorFlow框架的模型，执行模型导入（转换）时，其输出目录说明如下所示。

```

|
|---xxxx.om      转换输出的模型，可用于Ascend芯片，模型文件后缀统一为“.om”。
|---job_log.txt  转换过程的日志文件。
    
```

7.5 编写逻辑代码

华为HiLens提供了一套HiLens Framework，HiLens Framework通过封装底层接口、实现常用的管理功能，让开发者可以方便的开发Skill，发展AI生态。详细指导和API介绍可参见《开发指南》。

HiLens Framework 示例代码

HiLens Kit搭载了HiLens Framework，开发者在开发技能编写逻辑代码时，可以直接调用HiLens Framework的API。

下面是使用HiLens Framework获取摄像头内容并做简单处理的示例。

📖 说明

- 需要在开发技能时填写代码执行文件。
- 固件1.1.2及以后版本支持配置技能的Python依赖库。开发者在开发技能过程中，可根据自身需要配置技能的Python依赖库，详情请见[配置技能的Python依赖库](#)。

```

#!/usr/bin/python3
# skillframework 1.0.0 python demo

import hilens                                     # 导入hilens包
import cv2                                       # 导入opencv

def main():
    hilens.init("hello")                         # 技能初始化
    model = hilens.Model(hilens.get_model_dir() + # 构造模型管理器。
                          "faceDetection.om")   # 构造显示器
    disp = hilens.Display(hilens.HDMI)          # 构造摄像头
    cap = hilens.VideoCapture()                 # 构造预处理器
    proc = hilens.Preprocessor()

    hilens.set_log_level(hilens.DEBUG)          # 设定日志级别
    hilens.Info("This is a skillframework python demo") # 打印日志

    for i in range(10):
    
```

```

frame = cap.read() # 读取一帧画面
rframe = proc.resize(frame, 480, 480, 1) # 缩放
inputs = [rframe.flatten()] # 构造模型输入
outputs = model.infer(inputs) # 推理
hilens.Info("outputs len: %d" % len(outputs)) # 打印输出尺寸
for o in outputs:
    hilens.Info("output shape: " + str(o.shape))
# post process is ignored
disp.show(frame) # 将图像输出到显示器

hilens.terminate() # 销毁资源

if __name__ == "__main__":
    main()
    
```

7.6 编辑技能

在技能创建和使用过程中，如果发现其中信息需要纠正，您可以通过编辑操作，修改技能的基本信息、技能内容、运行时配置的信息。

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>技能管理”，进入技能列表。
2. 在“技能管理”页面，单击“操作”列的“编辑”，进入“编辑技能”页面。建议参考[使用空模板](#)相关的参数修改技能的内容。

图 7-18 编辑技能



7.7 安装和调试技能

技能创建完成后，您可以将技能安装至设备中，通过查看设备的技能视频或原始视频，查看技能运行效果，判断此技能是否满足业务诉求。

您还可以查看端侧日志调试技能，帮助定位问题，具体操作请参见[查看设备日志](#)。

前提条件

- 已有一个HiLens Kit设备，在华为HiLens中完成注册，且设备处于“在线”状态。设备管理相关指导请参见[设备管理简介](#)。
- 在华为HiLens中[新建技能](#)，技能已完成开发。

安装技能

1. 登录华为HiLens管理控制台，单击左侧导航栏“技能开发>技能管理”。进入“技能管理>基础技能”页面。
2. 选择需要安装的技能，单击右侧操作列“安装”。

图 7-19 安装技能



3. 在弹出的安装对话框中，勾选需要安装的设备，单击“确定”。当显示如下类似信息时，表示安装成功。

图 7-20 安装成功

安装技能“Gesture_Recognition”到设备。

设备名称	设备状态	资源约束	固件名称	固件版本	进度
<input checked="" type="radio"/> test	● 在线	<input checked="" type="radio"/> 路数 <input type="radio"/> 并发量	HiLens_Device_Ag...	1.0.9	✔ 安装成功
<input type="radio"/> test1	● 离线	<input checked="" type="radio"/> 路数 <input type="radio"/> 并发量	HiLens_Device_Ag...	1.0.7	-

4. 安装成功后，您可以前往[数据管理](#)，查看技能效果。

7.8 删除技能

如果您管理的技能将不再使用，您可以删除技能以释放资源。

1. 登录华为HiLens管理控制台，在左侧导航栏中选择“技能开发>技能管理”，进入技能列表。
2. 在“技能管理”页面，单击“操作”列的“删除”。弹出“删除技能”提示框。

图 7-21 删除技能



3. 确认删除信息后，在提示框中单击“确定”。

图 7-22 删除技能-23



📖 说明

删除后无法恢复, 请谨慎操作。

8 技能市场

8.1 技能市场简介

设备注册完成后，您可在华为HiLens平台的技能市场订购所需要的技能，拓展设备的AI能力。例如订购一个“人脸判断”技能，安装在设备上后，即可让设备具备判断人脸的能力。

图 8-1 技能市场



- 首先，您可以在“技能市场”中，通过筛选、搜索等操作，[查找您想要的技能](#)。
- 如果“技能市场”有您需要的技能，您可以直接[购买技能](#)。
- 对于您购买的技能，或者是您发布至市场的技能，您可以在“产品订购>订单管理”页面进行统一管理，详细操作请参见[管理订单](#)。

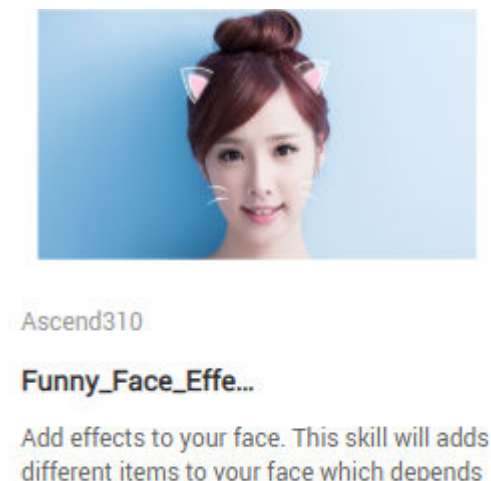
什么是技能

技能（Skill）是运行在端侧摄像头的人工智能应用，一般由模型和逻辑代码组成。其中，逻辑代码是技能的框架，负责控制技能的运行，包括数据读入、模型导入、模型推理、结果输出等；模型是人工智能算法经由大数据训练而成，负责技能运行中关键场景的推理。

华为HiLens的[技能市场](#)提供了丰富的技能供用户选择。

- 按应用场景划分，技能可应用于：智能园区、智慧家庭、智能车载、智能商超和其他等场景。
- 按不同的设备划分，技能分为2种，一种是适用于Ascend芯片的技能，另一种是适用于海思35XX系列芯片的技能。

图 8-2 技能卡片

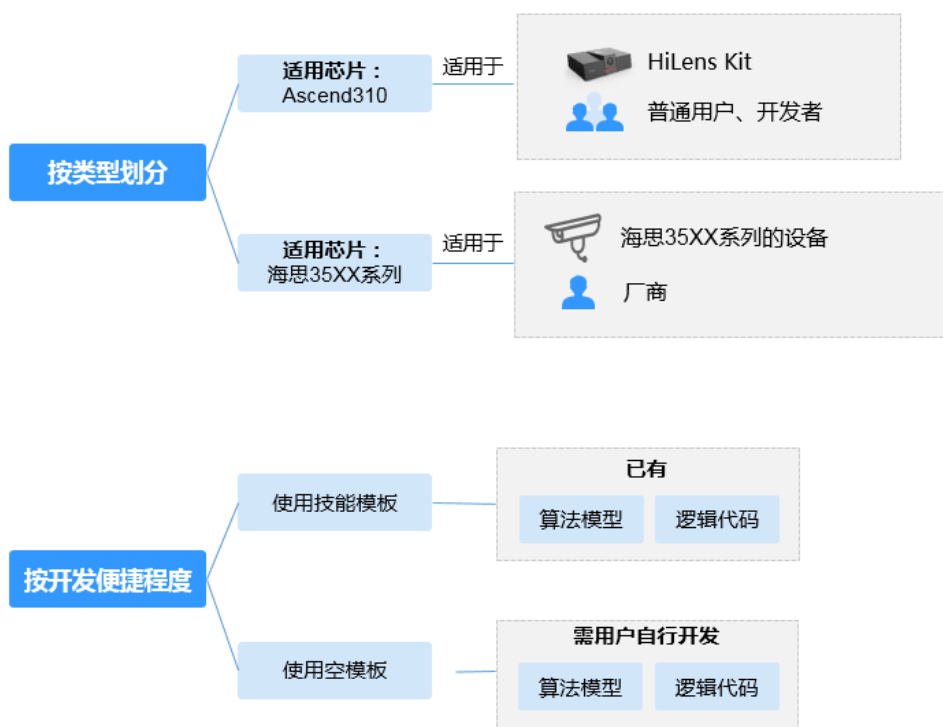


技能的分类

如下图所示，华为HiLens根据不同芯片，分为两种类型，分别可应用于HiLens Kit设备或其他海思35XX系列芯片的设备。平台不仅支持使用技能模块快速开发技能，便捷高效；还支持自行开发模型和逻辑代码，可以满足您更丰富的场景。

由于海思35XX系列芯片内存和性能偏低，针对适用于此类设备的技能需要对模型进行优化后才能运行，若需要开发此类技能，请联系华为工程师。

图 8-3 技能开发场景划分



8.2 在技能市场查找技能

由于技能市场是一个开放的市场，有预置技能，也有自己发布的技能，还有其他开发者分享的技能，在众多技能中，您可以通过如下几种方式查找您想要的技能。

技能市场默认按综合排序进行展示。

搜索技能

在“技能市场”页面左上角的“自定义过滤”中，输入技能名称的关键词，通过关键词进行搜索。

图 8-4 搜索



筛选技能

在已选条件下方，可以选择对应的“计费策略”、“发布方”、“适用芯片”、“OS”，筛选相关的技能，帮助您快速通过场景选择技能。其中，“发布方”分为“华为”和“第三方”，“华为”方指华为HiLens发布的官方技能，“第三方”指用户在华为HiLens控制台开发的技能并发布至技能市场的技能。

图 8-5 筛选技能



8.3 购买技能

不管是设备管理或者产品管理，首先您需要在“技能市场”购买一个可用的技能。

背景信息

- 购买技能需要一定的费用，在使用华为HiLens前检查账号状态，账号不能处于欠费或冻结状态。
- 技能可以分为“适用于Ascend芯片的技能”和“适用于海思35XX系列芯片的技能”，请仔细查看界面中技能的类型，然后根据实际情况进行购买。

在 HiLens 技能市场购买技能

1. 进入华为HiLens管理控制台，单击左侧导航栏的“产品订购>技能市场”，进入“技能市场”页面。
2. 在“技能市场”中，通过筛选、搜索等操作，找到想要安装的技能，单击技能图标进入“技能详情”页面。
3. 在技能详情页面，单击“立即购买”。
 - i. 在弹出的“购买技能”对话框中，按照“规格”选择“份数”，阅读技能隐私声明，并勾选“我同意该技能的隐私声明”，确认费用后，单击“立即购买”，完成技能购买。

当前“规格”为“路”，表示每份License可用在设备上的视频路数，有路数和并发量两种计量方式。如果以路数规格，一份License可以安装在设备上使用一路视频。

部分技能无需勾选“我同意该技能的隐私声明”，直接单击“立即购买”，即可完成技能购买。
 - ii. 技能购买完成后，对话框显示购买完成，您可以单击“订单”，返回“技能市场”页面。也可以在左侧导航栏中选择“产品订购>订单管理”进入“技能订单”页面，查看已购买技能。

图 8-6 购买技能



后续操作

- 针对普通用户或开发者（技能使用者）：您可以将购买的技能安装在设备上。详情请参见[安装技能](#)。
- 针对厂商（产品管理者）：您可以将购买的技能，分发至您的产品中。详情请参见[分发技能](#)。

8.4 管理订单

您可以统一管理在华为HiLens技能市场购买的技能。

技能订单

在华为HiLens管理控制台，选择左侧导航栏的“产品订购>订单管理”，默认进入“技能订单”页面。“技能订单”页面展示您购买的所有技能，以及购买时的详细信息。

在“技能订单”页面，您可以执行如下操作：

- **安装技能**

单击操作列的“安装”，将此技能安装至已注册的设备中，后续的详细操作指导可参见[安装技能](#)。

- **扩容**

- a. 单击购买详情列的“扩容”，增加购买的份数。
- b. 在“购买技能”页面，增加“购买数量”对应的数值，并勾选“我同意《声明》”，然后单击右下角的“立即购买”。
- c. 在“订单确认”页面确认订单详情，然后单击右下角的“提交订单”。
- d. 选择支付方式，然后单击右下角的“确认付款”，完成扩容。

仅包周期的计费技能支持扩容操作。

- **分发技能**

单击操作列的“分发”，将此技能分发至产品上，后续的详细操作可参见[分发技能](#)。

- **下载技能**

单击操作列的“下载”。浏览器自动下载技能的SDK包。

9 管理数据

当技能下发到设备端侧后，您可以在华为HiLens控制台下载设备的数据，便于事后回溯，也可以通过查看设备数据，查看技能运行效果。


📖 说明

- 数据管理里面只能管理HiLens Kit设备的数据。小海思芯片35xx系列的设备数据暂时不能在华为HiLens控制台管理。
- 只有以文件的方式输出到OBS供用户查看的技能，才能使用数据管理功能查看数据。

前提条件

- 普通用户成功将购买的技能安装到设备上，详情请参见[安装技能](#)。
- 开发者成功将技能部署到设备上，详情请参见[安装和调试技能](#)。
- 技能输出模块中定义了将数据输出至OBS的场景，针对此类技能，才能通过数据管理页面查看数据。
 - 如果您的技能开发者，您需要查看自己开发的技能在设备上的运行效果，在开发技能时，您需要指定数据存储至OBS中才能保证可以通过管理控制台下载。
 - 如果您是技能使用者，需要在技能市场中，通过技能详情页面了解此技能是否将数据存储至OBS中。

配置数据存储位置（OBS 存储路径）

1. 登录华为HiLens管理控制台，在管理控制台左侧菜单栏选择“设备管理 >设备列表”，然后在设备列表中，单击需要进行管理的某一设备，进入设备详情页面。
2. 在设备详情区域，您可以单击“数据存储位置”右侧的，然后在弹出对话框中选择您的OBS目录，设置当前设备的视频数据存储于您选择的OBS目录下。

📖 说明

- 由于设备数据的存储将耗费华为云的存储资源，使用OBS服务进行存储，将会产生相应的费用，详细操作及规范请参见[OBS控制台指南](#)。
- 由于通过URL访问桶时，桶名会作为URL的一部分，根据DNS标准，URL不支持大写字母，无法区分带大写字母的桶。因此，“桶名称”仅支持小写字母、数字、“-”、“.”作为桶的命名规则，禁止两个英文句号(.)或英文句号(.)和中划线(-)相邻。详细命名规则请参见[创建桶](#)。

图 9-1 配置数据存储位置



查看数据

您可根据数据输出类型，查看设备数据以及技能的运行效果，技能输出方式请见表 9-1。

说明

- 视频数据大多数技能通过设备的HDMI接口输出到显示屏，少数视频数据输出到服务器查看。请前往“技能市场”，在“技能详情”页面的“产品描述”中查看此技能的输出设置。例如，明星脸判断技能，其描述为“可以通过HDMI查看”，表示此技能部署后，需通过显示屏查看技能运行效果。
- 除“HDMI”输出方式以外，其他输出方式的技能存储在OBS中，需根据[配置数据存储位置（OBS存储路径）](#)查看数据。

表 9-1 技能输出方式

方式	查看数据说明
HDMI	直接通过设备的HDMI接口输出到显示屏。
RTMP	实时输出到服务器供用户查看，服务器地址请查看技能“产品描述”。
H264_FILE	以文件的方式输出到OBS供用户查看。 OBS文件路径请见 配置数据存储位置（OBS存储路径） 。

手动收集日志

当前设备固件版本在1.3.3及以后版本，需要手动收集日志。

1. 在华为HiLens管理控制台，单击左侧导航栏“数据管理（Beta）”，进入“数据管理”页面。
2. 单击选中日志上传失败的设备，然后单击右上角的“保存所有日志”。弹出“上传日志”对话框。

图 9-2 保存日志



3. 勾选需要上传到OBS的设备日志，单击“确定”。等待日志收集。

说明

日志按需收集，系统日志文件较大，收集耗时较长。

图 9-3 上传日志

上传日志

该操作将上传设备端的以下日志到OBS(对象存储服务)中，并产生一定的费用，详情请参考 [OBS计费](#)，上传前请确认已配置数据存储位置。若已存在日志文件夹，则将覆盖已有日志文件。

日志类型	描述
<input checked="" type="checkbox"/> 技能日志	安装在设备端的技能运行日志，记录技能运行状态、告警信息等日志。名称为技能名称_时间戳。
<input checked="" type="checkbox"/> HiLens Agent日志	HiLens管理日志，记录HiLens Agent管理的设备连接状态、技能安装状态、设备运行状态等日志，名称为HDADLOG。
<input checked="" type="checkbox"/> 厂商系统日志(仅限HiLens Kit)	系统软件、平台软件、NPU及系统应用等日志。格式为.tar.gz

确定

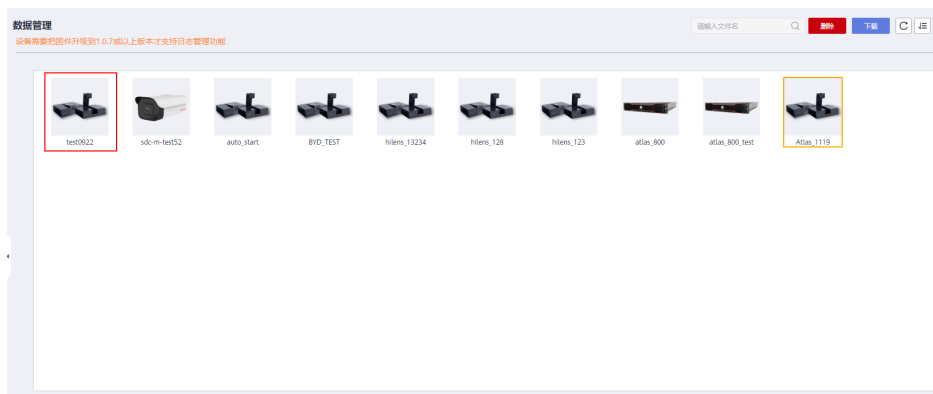
取消

4. 一般5分钟以后，待日志收集完成，在“数据管理”页面单击设备卡片，进入对应文件夹下载日志文件压缩包至本地，即可查看日志。

下载数据

1. 登录华为HiLens管理控制台，单击“数据管理(Beta)”，进入“数据管理”页面。
2. 在“数据管理”页面，双击指定的设备名称，进入待下载的设备数据文件夹。

图 9-4 数据管理

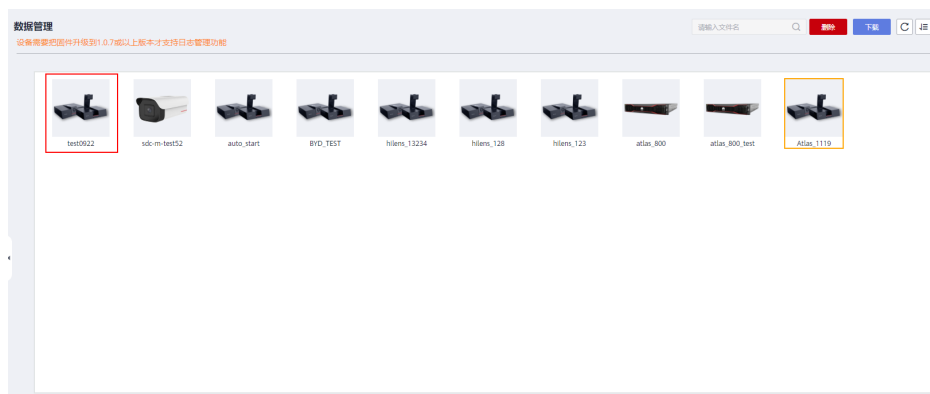


3. 选择需下载的数据文件夹，然后单击右上角的“下载”。将数据下载至本地。

删除数据

1. 登录华为HiLens管理控制台，单击“数据管理(Beta)”，进入“数据管理”页面。
2. 在“数据管理”页面，双击指定的设备名称，进入待删除的设备数据文件夹。

图 9-5 数据管理-41



3. 选择待删除的数据文件夹，然后单击右上角的“删除”，删除所选数据。

10 管理产品

10.1 产品简介

针对使用海思35xx系列芯片摄像头产品的厂商，您可以使用华为HiLens对您的设备产品进行批量管理，并且从技能市场订购技能，分发至您的产品中，使得您的设备具备AI能力。

- [新建产品](#)
- [订购技能（用于35XX系列设备）](#)
- [分发技能](#)
- [新增或删除技能](#)

10.2 新建产品

厂商在华为HiLens管理控制台创建一个产品后，将把设备“关联”到这个产品上来进行管理。购买设备的用户一旦将设备注册，厂商将能在华为HiLens控制台管理已注册的设备及其技能。

新建产品

1. 登录HiLens管理控制台，单击左侧导航栏“设备管理>产品管理”，进入“产品管理”页面。
2. 单击右上角“新建产品”，输入产品参数信息，如[表10-1](#)所示。

图 10-1 新建产品

产品名称

产品名称

* 平台

* 芯片

* 描述

表 10-1 设备产品参数

参数名称	参数说明
产品名称	产品名称。 请输入以字母开头，以字母、数字结尾，长度3~60的字符。只允许字母、数字、中划线和下划线等字符。
平台	设备的OS类型。 包括Android、Linux、iOS、LiteOS、Windows。
芯片	设备的芯片型号。 海思35XX系列，包括3516CV500、3519AV100、3519V101、3516DV300、3516EV200、3516EV300、3518EV300、ARM。 设备芯片型号为3518EV300\3516EV200\3516EV300时，均可选择“3516CV500”。
描述	产品介绍。 字符长度小于512。

- 产品信息确认无误后，单击“确定”。
新建成功后，系统将自动跳转至“产品管理”页面。

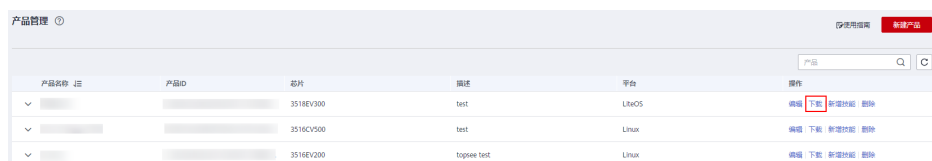
下载 License

产品新建成功后，您需要下载其对应的License，并存储至设备的指定路径。

- 单击左侧导航栏“设备管理>产品管理”，进入“产品管理”页面。

2. 选择对应的产品，然后单击操作列的“下载”，下载License文件。

图 10-2 下载 License



3. 将License文件存放至设备的指定目录下。
存放License的位置根据不同厂商的设备有所不同，与厂商达成合作关系后，您可以提交工单，会有华为HiLens工作人员专门指导。您也可以提交工单了解更多小海思35XX系列芯片的技能。

编辑产品

1. 登录HiLens管理控制台，单击左侧导航栏“设备管理>产品管理”，进入产品管理页面。
2. 选择对应的产品，单击操作列的“编辑”。
弹出“更新产品”对话框。

图 10-3 更新产品

产品名称

产品名称

* 平台

* 芯片

* 描述

3. 在“更新产品”对话框中，按表10-1编辑“产品名称”、“平台”、“芯片”和“描述”。
4. 确认产品信息，单击“确定”。

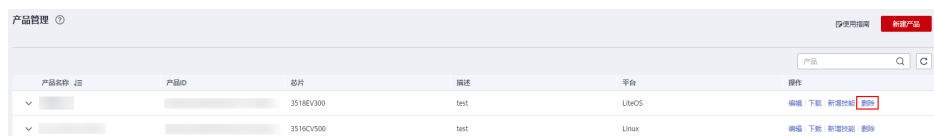
删除产品

须知

删除产品时需要确保此产品未投入使用，请谨慎操作。如果产品与设备关联，如果删除此产品将无法管理这些设备。如果用户已经注册过该设备，厂商将无法对这些设备分发技能。

1. 登录HiLens管理控制台，单击左侧导航栏“设备管理>产品管理”，进入产品管理页面。
2. 选择对应的产品，单击“删除”。

图 10-4 删除产品



3. 确认产品信息，单击“确定”。

10.3 订购技能（用于 35XX 系列设备）

您可在华为HiLens平台的技能市场订购所需要的技能，用于开发海思35XX系列的设备。技能适用的芯片包括：3516CV500、3519AV100、3519V101、3516DV300、3516EV200、3516EV300、3518EV300、ARM。

如果技能市场有您需要的技能，您可以直接购买技能。

- 购买技能：在技能市场中，您需要选择“适用于海思35XX系列芯片”的技能。详细指导请参见[购买技能](#)。

10.4 分发技能

“适用于海思35XX系列芯片的技能”需要分发至设备上才能使用。针对使用海思35xx系列芯片的设备，由于性能限制，最多只能分发5个技能。

前提条件

- 已将设备管理至产品，参见[新建产品](#)。
- 已订购技能，参见[订购技能（用于35XX系列设备）](#)。

分发技能

1. 登录华为HiLens管理控制台，单击左侧导航栏“产品订购>订单管理”，默认进入“技能订单”页面。
2. 在“技能订单”页面，选择对应的技能，将技能的License分发至产品上。技能信息中，“适用芯片”不是“Ascend 310”的技能才能分发到产品。
 - a. 选择需要分发的技能，单击“操作”列的“分发”。

图 10-5 分发技能



- b. 在弹出的“分发到产品”的对话框中，勾选需要分发的产品，根据自身需要选择勾选“自动升级”的“是”，然后单击“分发到产品”。
- 勾选“自动升级”表示如果分发的技能有新的版本，产品中的该技能会自动升级到最新版本。

图 10-6 分发技能至产品



- c. 对话框内容显示技能分发情况，确认“安装状态”为“成功”，单击“确定”完成技能分发操作。
3. 下载技能。
- 在“技能订单”页面，在分发的技能中选择需要使用的技能，单击操作列的“下载”。浏览器自动下载技能的SDK包。

图 10-7 下载技能



4. 部署技能。将步骤3下载的SDK包集成到设备中。
- 集成到设备的具体位置根据不同的设备有所差异，操作指导请联系华为工程师。

10.5 新增或删除技能

分发技能是指从技能市场，将技能分发至产品设备；而新增技能是指在产品管理页面，为指定产品新增技能。两个操作的最终结果是一样的。

在当前产品中，如果某些技能不再使用，可将其删除。

新增技能

1. 登录华为HiLens管理控制台，单击左侧导航栏“设备管理>产品管理”，进入“产品管理”页面。
2. 选择设备产品，单击操作列的“新增技能”。
3. 在弹出的“新增技能”窗口，默认展示您订购的技能列表，选择需要新增的技能，根据需要勾选“自动升级”，单击“确定”完成操作。

勾选“自动升级”表示如果分发的技能有新的版本，产品中的该技能会自动升级到最新版本。

图 10-8 新增技能

可用技能

<input type="checkbox"/> 技能名称	平台	份数	购买时间	自动升级
<input type="checkbox"/> test-privacy-protecti...	Linux	1	2021/07/27...	<input type="checkbox"/>
<input type="checkbox"/> test-privacy-protecti...	Linux	1	2021/07/10...	<input type="checkbox"/>
<input type="checkbox"/> test-privacy-protecti...	Linux	1	2021/07/10...	<input type="checkbox"/>
<input type="checkbox"/> Human_Face_Recog...	Linux	1	2021/06/28...	<input type="checkbox"/>
<input type="checkbox"/> Human_Face_Recog...	Linux	3	2021/06/23...	<input type="checkbox"/>

5 总条数: 101 < 1 2 3 4 5 ... 21 >

确定
取消

删除技能

删除技能是将云侧的技能从设备中移除。

说明

- 对于支持自动更新的设备（使用3516cv500、3519av100芯片的设备），检测到云侧的技能删除后，会将端侧的技能删掉。
- 对于不支持自动更新的设备，云侧技能删除后，端侧不受影响，依然可以使用技能。若希望删除端侧的技能，您可以提交工单。


1. 登录华为HiLens管理控制台，单击左侧导航栏“设备管理>产品管理”，进入“产品管理”页面。
2. 选择设备产品，单击左侧箭头 ，查看产品下的所有技能。
3. 选择需要删除的技能，单击“删除”。

图 10-9 删除技能-42

产品名称	产品ID	芯片	描述	平台	操作
产品3518	bb498cd8e9425db797174f18f0e...	3518EV300	test	LiteOS	编辑 下载 新建技能 删除

技能名称	版本	更新时间	自动升级	操作
test-privacy-protection	0.0.1	2021/04/12 20:42:35 GMT+08:00	<input type="checkbox"/>	删除
Human_Face_Recognition_3518EV200_HiLinux_TOPSEE	0.1.2	2021/03/27 17:43:24 GMT+08:00	<input type="checkbox"/>	删除
KeywordSpotting_Android	1.0.2	2020/09/11 15:13:01 GMT+08:00	<input type="checkbox"/>	删除

5 总数: 3 < 1 >

4. 确认需删除的技能信息，单击“确定”。

A Caffe 算子边界

对于Caffe框架，当算子的输入维度不是4时，如果存在axis参数，不能使用负数。
“.om”模型支持的Caffe算子边界如表A-1所示。

表 A-1 Caffe 算子边界

序号	算子	含义	边界
1	Absval	对输入求绝对值	<p>【输入】 1个输入</p> <p>【参数】 engine: 枚举型，默认为0 (DEFAULT=0, CAFFE=1, CUDNN=2)，可选</p> <p>【约束】 无限制</p> <p>【量化工具支持】 是</p>
2	Argmax	返回输入的最大值对应的索引序号	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • out_max_val: 布尔型，默认为false，可选 • top_k: unit32，默认为1，可选 • axis: int32，可选 <p>【约束】 无限制</p> <p>【量化工具支持】 否</p>

序号	算子	含义	边界
3	BatchNorm	对输入做标准化 $(x - \text{avg}(x)) / x$ 的标准差	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • use_global_stats: 布尔型, 必须为 true • moving_average_fraction: float, 默认为0.999, 可选 • eps: float, 默认为1e-5, 可选 <p>【约束】 仅支持对C维度方向做归一化</p> <p>【量化工具支持】 是</p>
4	Concat	输入数据按维度拼接	<p>【输入】 多个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • concat_dim: uint32, 默认为1, 取值: 大于0, 可选 • axis: int32, 默认1, 可选; 与 concat_dim功能相同, 二者择其一; axis==-1时, 输入维度必须等于4, 否则结果可能错误 <p>【约束】</p> <ul style="list-style-type: none"> • 输入的tensor, 除了进行concat的维度外, 其他维度的size必须相等 • 输入tensor的个数范围不超过[1, 1000] <p>【量化工具支持】 是</p>

序号	算子	含义	边界
5	DepthwiseConvolution	深度卷积	<p>【输入】 1个输入，且filter为常量，维度为4</p> <p>【参数】</p> <ul style="list-style-type: none"> • num_output: uint32, 可选 • bias_term: 布尔型，默认为true, 可选 • pad: uint32, 默认为0; 数组 • kernel_size: uint32; 数组 • stride: uint32, 默认为1; 数组 • dilation: uint32, 默认为1; 数组 • pad_h: uint32, 默认为0, 可选 (仅2D) • pad_w: uint32, 默认为0, 可选 (仅2D) • kernel_h: uint32, 可选 (仅2D) • kernel_w: uint32, 可选 (仅2D) • stride_h: uint32, 可选 (仅2D) • stride_w: uint32, 可选 (仅2D) • group: uint32, 默认为1, 可选 • weight_filler: 类型为FillerParameter, 可选 • bias_filler: 类型为FillerParameter, 可选 • engine: 枚举型，默认为0 (DEFAULT=0, CAFFE=1, CUDNN=2), 可选 • force_nd_im2col: 布尔型，默认为false, 可选 • axis: int32, 默认为1, 可选 <p>【约束】 filterN=inputC=group</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
6	Convolution	卷积	<p>【输入】 1个输入，且filter为常量，维度为4</p> <p>【参数】</p> <ul style="list-style-type: none"> • num_output: uint32, 可选 • bias_term: 布尔型，默认为true, 可选 • pad: uint32, 默认为0; 数组 • kernel_size: uint32; 数组 • stride: uint32, 默认为1; 数组 • dilation: uint32, 默认为1; 数组 • pad_h: uint32, 默认为0, 可选 (仅2D) • pad_w: uint32, 默认为0, 可选 (仅2D) • kernel_h: uint32, 可选 (仅2D) • kernel_w: uint32, 可选 (仅2D) • stride_h: uint32, 可选 (仅2D) • stride_w: uint32, 可选 (仅2D) • group: uint32, 默认为1, 可选 • weight_filler: 类型为FillerParameter, 可选 • bias_filler: 类型为FillerParameter, 可选 • engine: 枚举型，默认为0 (DEFAULT=0, CAFFE=1, CUDNN=2), 可选 • force_nd_im2col: 布尔型，默认为false, 可选 • axis: int32, 默认为1, 可选 <p>【约束】</p> <ul style="list-style-type: none"> • $(inputW + padWHead + padWTail) \geq ((FilterW - 1) * dilationW) + 1$ • $(inputW + padWHead + padWTail) / StrideW + 1 \leq 2147483647$ • $(inputH + padHHead + padHTail) \geq ((FilterH - 1) * dilationH) + 1$

序号	算子	含义	边界
			<ul style="list-style-type: none"> • $(inputH + padHHead + padHTail) / StrideH + 1 \leq 2147483647$ • $0 \leq Pad < 256, 0 < FilterSize < 256, 0 < Stride < 64, 1 \leq dilationsize < 256$ <p>【量化工具支持】 是</p>
7	Crop	截取	<p>【输入】 2个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • axis: int32, 默认为2, axis=-1, input dim必须等于4, 可选 • offset: uint32, 数组 <p>【约束】 无限制</p> <p>【量化工具支持】 否</p>

序号	算子	含义	边界
8	Deconvolution	反卷积	<p>【输入】 1个输入，且filter为常量，维度为4</p> <p>【参数】</p> <ul style="list-style-type: none"> • num_output: uint32, 可选 • bias_term: 布尔型，默认为true, 可选 • pad: uint32, 默认为0; 数组 • kernel_size: uint32; 数组 • stride: uint32, 默认为1; 数组 • dilation: uint32, 默认为1; 数组 • pad_h: uint32, 默认为0, 可选 (仅2D) • pad_w: uint32, 默认为0, 可选 (仅2D) • kernel_h: uint32, 可选 (仅2D) • kernel_w: uint32, 可选 (仅2D) • stride_h: uint32, 可选 (仅2D) • stride_w: uint32, 可选 (仅2D) • group: uint32, 默认为1, 可选 • weight_filler: 类型为FillerParameter, 可选 • bias_filler: 类型为FillerParameter, 可选 • engine: 枚举型，默认为0 (DEFAULT=0, CAFFE=1, CUDNN=2), 可选 • force_nd_im2col: 布尔型，默认为false, 可选 • axis: int32, 默认为1, 可选 <p>【约束】</p> <ul style="list-style-type: none"> • group = 1 • dilation = 1 • filterH - padHHead - 1 >= 0 • filterW - padWHead - 1 >= 0 <p>还有一条约束涉及中间变量，公式如下：</p> <ol style="list-style-type: none"> 1. $a = \text{ALIGN}(\text{filter_num}, 16) * \text{ALIGN}(\text{filter_c}, 16) * \text{filter_h} * \text{filter_w} * 2$ 2. 如果$\text{ALIGN}(\text{filter_c}, 16) \% 32 = 0$, $a = a/2$

序号	算子	含义	边界
			3. $\text{conv_input_width} = (\text{反卷积输入}w - 1) * \text{strideW} + 1$ 4. $b = (\text{conv_input_width}) * \text{filter_h} * \text{ALIGN}(\text{filter_num}, 16) * 2 * 2$ 5. $a + b \leq 1024 * 1024$ 【量化工具支持】 是

序号	算子	含义	边界
9	Detection Output	检测结果输出FSR	<p>【输入】 3个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • num_classes: 必选, 类型: int32, 要预测的类数 • share_location: 可选, 类型: bool, 默认为true (表示不同类间共享bounding box) • background_label_id: 可选, 类型: int32, 默认为0 • nms_param: 可选, 非最大抑制 • save_output_param: 可选, 用于保存检测结果 • code_type: 可选, 默认为CENTER_SIZE • variance_encoded_in_target: 可选, 类型: bool, 默认为true; 如果为true, 方差编码在目标中, 否则需要相应地调整预测偏移量 • keep_top_k: 可选, 类型: int32, 在nms步骤后每个图像要保留的总bbox数 • confidence_threshold: 可选, 类型: float, 仅考虑置信度大于阈值的检测; 如果没有设置, 考虑所有的box • nms_threshold: 可选, 类型: float • top_k: 可选, 类型: int32 • boxes: 可选, 类型: int32, 默认为1 • relative: 可选, 类型: bool, 默认为true • objectness_threshold, 可选, 类型: float, 默认为0.5 • class_threshold: 可选, 类型: float, 默认为0.5 • biases: 数组 • general_nms_param: 可选 <p>【约束】</p> <ul style="list-style-type: none"> • 支持fasterrcnn网络 • 非极大值抑制比nmsThreshold $\in (0, 1)$

序号	算子	含义	边界
			<ul style="list-style-type: none"> • 概率阈值postConfThreshold\in (0, 1) • 类别≥ 2 • 最大支持1024个框输入 • 输出的w维度是16 【量化工具支持】 是
10	Eltwise	按元素操作层（求和、乘积、最大值）	【输入】 至少2个输入 【参数】 <ul style="list-style-type: none"> • operation: 可选，类型：枚举型（PROD = 0; SUM = 1; MAX = 2），默认为SUM • coeff: 数组；类型：float • stable_prod_grad: 可选，类型：bool，默认为true 【约束】 <ul style="list-style-type: none"> • 最大支持4个输入 • 与原生算子相比，不支持stable_prod_grad参数 • 支持prod、sum、max三种模式 【量化工具支持】 是
11	Elu	激活函数	【输入】 1个输入 【参数】 alpha (optional) : 类型：float，默认为1 【约束】 无限制 【量化工具支持】 否

序号	算子	含义	边界
12	Exp	对输入求以e为底的x次方值	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • base: 可选, 类型: float, 默认为-1.0 • scale: 可选, 类型: float, 默认为1.0 • shift: 可选, 类型: float, 默认为0.0 <p>【约束】 无限制</p> <p>【量化工具支持】 否</p>
13	Flatten	输入 $n*c*h*w$ 变为向量 $n*(c*h*w)$	<p>【输入】 1个输入 (top_size与bottom_size必须不一致, 且不为1; 当axis=-1时, 输入的dim必须等于4)</p> <p>【参数】</p> <ul style="list-style-type: none"> • axis: 可选, 类型: int32, 默认为1 • end_axis: 可选, 类型: int32, 默认为-1 <p>【约束】 axis必须小于end axis</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
14	FullConnection	全连接	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • num_output: 可选, 类型: uint32 • bias_term: 可选, 类型: bool, 默认为true • weight_filler: 可选, 类型: FillerParameter, 维度为2 • bias_filler: 可选, 类型: FillerParameter, 维度为1 • axis: 可选, 类型: int32, 默认为1 • transpose: 可选, 类型: bool, 默认为false <p>【约束】</p> <ul style="list-style-type: none"> • 仅支持transpose=false, axis=1 • Bias_C <= 56832 • 如果客户要量化模型时, 需要满足下列维度: <ul style="list-style-type: none"> - 当N = 1, $2 * \text{CEIL}(C, 16) * 16 * xH * xW \leq 1024 * 1024$ - 当N > 1, $2 * 16 * \text{CEIL}(C, 16) * 16 * xH * xW \leq 1024 * 1024$ <p>【量化工具支持】 是</p>

序号	算子	含义	边界
15	Interp	插值层	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • height: 可选, 类型: int32, 默认为0 • width: 可选, 类型: int32, 默认为0 • zoom_factor: 可选, 类型: int32, 默认为1 • shrink_factor: 可选, 类型: int32, 默认为1 • pad_beg: 可选, 类型: int32, 默认为0 • pad_end: 可选, 类型: int32, 默认为0 <p>说明</p> <ul style="list-style-type: none"> l zoom_factor与shrink_factor不能同时存在 l height与zoom_factor不能同时存在 l height与shrink_factor不能同时存在 <p>【约束】 $(outputH * outputW) / (inputH * inputW) > 1/30$</p> <p>【量化工具支持】 否</p>
16	Log	对输入进行log计算	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • base: 可选, 类型: float, 默认为-1.0 • scale: 可选, 类型: float, 默认为1.0 • shift: 可选, 类型: float, 默认为0.0 <p>【约束】 无限制</p> <p>【量化工具支持】 否</p>

序号	算子	含义	边界
17	LRN	局部响应归一化层	<p>【输入】 1个输入，不支持常量输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • local_size: 可选，类型：uint32，默认为5 • alpha: 可选，类型：float，默认为1 • beta: 可选，类型：float，默认为0.75 • norm_region: 可选，类型：枚举型，默认为ACROSS_CHANNELS，(ACROSS_CHANNELS=0; WITHIN_CHANNEL = 1) • lrnk: 可选，类型：float，默认为1 • engine: 可选，类型：枚举型 (DEFAULT = 0; CAFFE = 1; CUDNN = 2) <p>【约束】</p> <ul style="list-style-type: none"> • local_size >0，且必须为奇数 • 通道间：当local_size ∈ [1, 15]时，lrnk>0.00001且beta>0.01，否则lrnk和beta为任意值；Lrnk和alpha不同时为0；当C维度大于1776时，local_size <1728 • 通道内：Lrnk=1，local_size ∈ [1, 15]，beta>0.01 <p>【量化工具支持】 是</p>

序号	算子	含义	边界
18	LSTM	长短期记忆网络	<p>【输入】 2个或3个输入</p> <ul style="list-style-type: none"> • X: 时间序列数据(T*B*Xt), 按4D的NCHW格式排布传入 • 需要保证: N对应时间序列长度T, C对应Batch数B, H对应t时刻的输入数据Xt, W取1 • Cont: 序列连续性标志 (T*B) • Xs: 静态数据 (B*Xt), 可选 <p>【参数】</p> <ul style="list-style-type: none"> • num_output: 可选, 类型: uint32, 默认为0 • weight_filler: 可选, 类型: FillerParameter • bias_filler: 可选, 类型: FillerParameter • debug_info: 可选, 类型: bool, 默认为false • expose_hidden: 可选, 类型: bool, 默认为false <p>【约束】 此约束涉及中间变量计算, 公式如列: $a = (\text{ALIGN} (xt, 16) + \text{ALIGN} (\text{output}, 16)) * 16 * 2 * 2$ $b = (\text{ALIGN} (xt, 16) + \text{ALIGN} (\text{output}, 16)) * 16 * 4 * 2 * 2$ $c = \text{use_projection} ? \text{ALIGN} (ht, 16) * \text{ALIGN} (\text{output}, 16) * 2 : 0$ $d = 16 * \text{ALIGN} (ht, 16) * 2$ $e = \text{batchNum} * 4$ 则, 约束为: $a + b + c \leq 1024 * 1024$ $d \leq 256 * 1024 / 8$ $e \leq 256 * 1024 / 32$ </p> <p>【量化工具支持】 否</p>

序号	算子	含义	边界
19	Normalize	标准化层	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> across_spatial: 可选, 类型: bool, 默认为true scale_filler: 可选, 默认为1.0 channel_shared: 可选, 类型: bool, 默认为true eps: 可选, 类型: float, 默认为1e-10 <p>【约束】</p> <ul style="list-style-type: none"> eps应大于1e-7, 并且小于等于0.1+ (1e-6) Caffe框架中的参数across_spatial目前只支持true, 按channel进行norm操作 <p>【量化工具支持】 是</p>
20	Permute	将输入维度按照给定模式进行重排	<p>【输入】 1个输入</p> <p>【参数】 order: uint32, 数组</p> <p>【约束】 无约束</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
21	Pooling	池化层	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> pool: 池化方法, 可选, 类型: 枚举, 取值: MAX=0, AVE=1, STOCHASTIC = 2, 默认为MAX pad: 可选, 类型: uint32, 默认为0 pad_h: 可选, 类型: uint32, 默认为0 pad_w: 可选, 类型: uint32, 默认为0 kernel_size: 可选, 类型: uint32, kernel_size和kernel_h/kernel_w不能同时出现 kernel_h: 可选, 类型: uint32 kernel_w: 可选, 类型: uint32, kernel_h/kernel_w必须同时存在 stride: 可选, 类型: uint32, 默认为1 stride_h: 可选, 类型: uint32 stride_w: 可选, 类型: uint32 engine: 可选, 类型: 枚举, 取值: DEFAULT=0, CAFFE=1, CUDNN=2 global_pooling: 可选, 类型: bool, 默认值为false ceil_mode: 可选, 类型: bool, 默认为true round_mode: 可选, 类型: 枚举, 取值: CEIL=0, FLOOR=1; 默认为CEIL <p>【约束】</p> <ul style="list-style-type: none"> kernelH<=inputH+padTop+padBottom kernelW<=inputW+padLeft+padRight padTop<windowH padBottom<windowH padLeft<windowW padRight<windowW

序号	算子	含义	边界
			<p>除公共约束外，还需满足下列某一场景支持范围</p> <p>global pool模式只支持以下范围：</p> <ol style="list-style-type: none"> outputH==1 && outputW==1 && kernelH>=inputH && kernelW>=inputW inputH*inputW<=10000 <p>【量化工具支持】 是</p>
22	Power	计算 $y = (scale * x + shift) ^ power$	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> power: 可选，类型：float，默认为1.0 scale: 可选，类型：float，默认为1.0 shift: 可选，类型：float，默认为0.0 <p>【约束】</p> <ul style="list-style-type: none"> power!=1 scale*x+shift>0 <p>【量化工具支持】 是</p>
23	Prelu	激活函数	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> filler: 可选 channel_shared: 可选，类型：bool，默认为false，是否跨channel共享斜率参数 <p>【约束】 无限制</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
24	PriorBox	从目标预选框获取真实目标的位置	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • min_size: 必须设置, 最小框大小 (以像素为单位) • max_size: 必须设置, 最大框大小 (以像素为单位) • aspect_ratio: 数组; 类型: float; 各种宽高比, 重复的比率将被忽略, 如果没有提供, 使用默认比率1 • flip: 可选, 类型: bool, 默认为 true; 如果为 true, 将翻转每个宽高比, 例如, 如果有宽高比 'r', 也将生成宽高比 '1.0/r' • clip: 可选, 类型: bool, 默认为 false; 如果为 true, 将剪切先前的值, 使其在 [0, 1] 范围内 • variance: 数组; 调整先前 bbox 的方差 • img_size: 可选, 类型: uint32, img_size 与 img_h/img_w 不能同时存在 • img_h: 可选, 类型: uint32 • img_w: 可选, 类型: uint32 • step: 可选, 类型: float, step 与 step_h/step_w 不能同时存在 • step_h: 可选, 类型: float • step_w: 可选, 类型: float • offset: 类型: float, 默认为 0.5 <p>【约束】 只支持 ssd 网络 输出维度: [n, 2, 检测框*4, 1]</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
25	Proposal	将预选框通过 (proposal, score) 排序, 通过 nms 获取 topN proposal	<p>【输入】 3个输入 (scores, bbox_pred, im_info)</p> <p>【参数】</p> <ul style="list-style-type: none"> • feat_stride: 可选, 类型: float • base_size: 可选, 类型: float • min_size: 可选, 类型: float • ratio: 数组; 类型: float • scale: 数组; 类型: float • pre_nms_topn: 可选, 类型: int32 • post_nms_topn: 可选, 类型: int32 • nms_thresh: 可选, 类型: float <p>【约束】</p> <ul style="list-style-type: none"> • 只用于fastercnn • ProposalParameter、PythonParameter不能同时存在 • preTopK范围为1~6144 • postTopK范围为1~1024 • scaleCnt*ratioCnt的最大值支持到64 • nmsTresh: 过滤框使用的阈值, $0 < nmsTresh \leq 1$ • minSize: 框的边长的最小值, 所有小于此最小值的框将会被过滤掉 • featStride: 在生成默认框时, 指定两个相邻的框延H或W的步长 • baseSize: 用来生成默认框用到的参数, 表示框的基本大小 • ratio & scale: 生成默认框用到的参数 • imgH&imgW: 输入到网络的图片高和宽, 其值必须大于0 • input维度约束: clsProb: $C=2*scaleCnt*ratioCnt$ bboxPred: $C=4*scaleCnt*ratioCnt$ bboxPrior: $N=clsProb.N, C=4*scaleCnt*ratioCnt$ imInfo: $N=clsProb.N, C=3$ <p>【量化工具支持】</p>

序号	算子	含义	边界
			是
26	PSROI Pooling	位置敏感的区域池化	<p>【输入】 2个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • spatial_scale: 必须配置, 类型: float • output_dim: 必须配置, 类型: int32, 输出通道数 • group_size: 必须配置, 类型: int32, 编码位置敏感分数图的组数 <p>【约束】 用于RFCN网络</p> <ul style="list-style-type: none"> • 输入Roi框的坐标信息为[roiN, roiC, roiH, roiW], 格式范围是: $1 \leq \text{roiN} \leq 65535$, $\text{roiC} == 5$, $\text{roiH} == 1$, $\text{roiW} == 1$ • 输入的featuremap维度为[xN, xC, xH, xW] $\text{pooledH} == \text{pooledW} == \text{groupSize} \leq 128$ [pooledH pooledW]表示pool框的长宽 • 输出的格式 y[yN, yC, yH, yW] • poolingMode == avg pooling, $\text{pooledH} == \text{pooledW} == \text{groupSize}$, $\text{pooledH} \leq 128$, $\text{spatialScale} > 0$, $\text{groupSize} > 0$, $\text{outputDim} > 0$ • $1 \leq \text{xN} \leq 65535$, $\text{roisN} \% \text{xN} == 0$ • HW_LIMIT是xh、xw的约束 $\text{xHW} = \text{xH} * \text{xW}$ $\text{pooledHW} = \text{pooledH} * \text{pooledW}$ $\text{HW_LIMIT} = (64 * 1024 - 8 * 1024) / 32$ $\text{xH} \geq \text{pooledH}$, $\text{xW} \geq \text{pooledW}$ $\text{xHW} \geq \text{pooledHW}$ $\text{xHW} / \text{pooledHW} \leq \text{HW_LIMIT}$ • 多batch场景时, 每个batch的roi框个数相同, 且roi的batch排列顺序与feature相同 <p>【量化工具支持】 是</p>

序号	算子	含义	边界
27	Relu	激活函数，同时包含普通的relu和leaky relu，可通过参数指定	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> negative_slope: 可选，类型：float，默认为0 engine: 可选，类型：枚举，取值：DEFAULT=0, CAFFE=1, CUDNN=2 <p>【约束】 无限制</p> <p>【量化工具支持】 是</p>
28	Reshape	改变输入维度	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> shape: 常量，类型：int64或int axis: 可选，类型：int32，默认为0 num_axes: 可选，类型：int32，默认为-1 <p>【约束】 无限制</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
29	ROIAlign	一种区域特征聚集的方式	<p>【输入】 至少有2个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • pooled_h: 可选, 类型: uint32, 默认为0 • pooled_w: 可选, 类型: uint32, 默认为0 • spatial_scale: 可选, 类型: float, 默认为1 • sampling_ratio: 可选, 类型: int32, 默认为-1 <p>【约束】 主要用于maskrcnn</p> <ul style="list-style-type: none"> • FeatureMap (特征图) 约束: <ul style="list-style-type: none"> (1) $H * W \leq 5248 (N > 1)$ 或 $W * C < 40960 (N = 1)$ (2) $C \leq 1280$ (3) $((C - 1) / 128 + 1) * pooledW \leq 216$ • Roi (感兴趣区域) 约束: <ul style="list-style-type: none"> (1) $C = 5$ (caffe), $H = 1, W = 1$ (2) $samplingRatio * pooledW \leq 128$ 且 $samplingRatio * pooledH \leq 128$ (3) $H \geq pooledH, W \geq pooledW$ <p>【量化工具支持】 是</p>

序号	算子	含义	边界
30	ROI Pooling	将“候选框”映射到特征图上	<p>【输入】 至少有2个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • pooled_h: 必须设置, 类型: uint32, 默认为0 • pooled_w: 必须设置, 类型: uint32, 默认为0 • spatial_scale: 必须设置, 类型: float; 默认为1; 乘法空间比例因子将ROI坐标从其输入比例转换为pool时使用的比例 <p>【约束】 主要用于fasterrcnn</p> <ul style="list-style-type: none"> • 输入维度: $H * W \leq 8160$; 且 $H \leq 120$; 且 $W \leq 120$ • 输出维度: $pooledH \leq 20$, 且 $pooledW \leq 20$ <p>【量化工具支持】 是</p>
31	Scale	$out = \alpha * Input + \beta$	<p>【输入】 2个输入, input的dim为4</p> <p>【参数】</p> <ul style="list-style-type: none"> • axis: 可选, 类型: int32, 默认为1, 仅支持axis为1或者-3 • num_axes: 可选, 类型: int32, 默认为1 • filler: 可选; filler被忽略, 除非只给1个bottom且scale是学习参数 • bias_term: 可选, 类型: bool, 默认为false; 是否也学习bias (相当于ScaleLayer + BiasLayer, 但可能更有效率), 使用bias_filler初始化 • bias_filler: 可选, 默认为0 <p>【约束】 scale, bias的shape只支持 (n, c, 1, 1), 且c维度与input的c维度相等</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
32	ShuffleChannel	帮助信息在特征通道交叉流动	<p>【输入】 1个输入</p> <p>【参数】 group: 可选, 类型: uint32, 默认为1</p> <p>【约束】 无限制</p> <p>【量化工具支持】 是</p>
33	Sigmoid	激活函数	<p>【输入】 1个输入</p> <p>【参数】 engine: 可选, 类型: 枚举, 取值: DEFAULT=0, CAFFE=1, CUDNN=2</p> <p>【约束】 无限制</p> <p>【量化工具支持】 是</p>
34	Slice	将输入分解成多个输出	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • slice_dim: 可选, 类型: uint32, 默认为1; axis和slice_dim不能同时存在 • slice_point: 数组; 类型: uint32 • axis: 可选, 类型: int32, 默认为1 (表示沿channel拼接) <p>【约束】 无限制</p> <p>【输出】 无限制</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
35	Softmax	归一化逻辑函数	<p>【输入】 1个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> engine: 可选, 取值: DEFAULT=0, CAFFE=1, CUDNN=2 axis: 可选, 类型: int32, 默认为1; 表示沿哪个axis作softmax <p>【约束】 输入4维时可以针对每一维做softmax: 根据分类的轴不同, 计算边界分别为:</p> <ul style="list-style-type: none"> axis=1时, $c \leq ((256 \times 1024 / 4) - 8 \times 1024 - 256) / 2$ axis=0时, $n \leq (56 \times 1024 - 256) / 2$ axis=2时, $W=1, 0 < h < (1024 \times 1024 / 32)$ axis=3时, $0 < W < (1024 \times 1024 / 32)$ <p>输入维度不足4维时, 仅支持对最后一维做softmax计算, 并且最后一维不超过46080</p> <p>【量化工具支持】 是</p>
36	Tanh	激活函数	<p>【输入】 1个输入</p> <p>【参数】 engine: 可选, 类型: 枚举, 取值: DEFAULT=0, CAFFE=1, CUDNN=2</p> <p>【约束】 输入tensor元素个数不超过INT32_MAX</p> <p>【量化工具支持】 是</p>
37	Upsample	maxpool的反向传播过程	<p>【输入】 2个输入</p> <p>【参数】 scale: 可选, 类型: int32, 默认为1</p> <p>【约束】 无限制</p> <p>【量化工具支持】 是</p>

序号	算子	含义	边界
38	SSDDetectionOutput	SSD网络检测输出	<p>【输入】 3个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • num_classes: 必选, 类型: int32, 要预测的类数 • share_location: 可选, 类型: bool, 默认为true (表示不同类间共享bounding box) • background_label_id: 可选, 类型: int32, 默认为0 • nms_param: 可选, 非最大抑制 • save_output_param: 可选, 用于保存检测结果 • code_type: 可选, 默认为CENTER_SIZE • variance_encoded_in_target: 可选, 类型: bool, 默认为true; 如果为true, 方差编码在目标中, 否则需要相应地调整预测偏移量 • keep_top_k: 可选, 类型: int32, 在nms步骤后每个图像要保留的总bbox数 • confidence_threshold: 可选, 类型: float, 仅考虑置信度大于阈值的检测; 如果没有设置, 考虑所有的box • nms_threshold: 可选, 类型: float • top_k: 可选, 类型: int32 • boxes: 可选, 类型: int32, 默认为1 • relative: 可选, 类型: bool, 默认为true • objectness_threshold, 可选, 类型: float, 默认为0.5 • class_threshold: 可选, 类型: float, 默认为0.5 • biases: 数组 • general_nms_param: 可选 <p>【约束】</p> <ul style="list-style-type: none"> • 只支持SSD网络 • preTopK和postTopK的取值范围当前仅支持1~1024

序号	算子	含义	边界
			<ul style="list-style-type: none"> • shareLocation仅支持true • nmsEta仅支持1 • numClasses支持的范围是1~2048 • code_type仅支持CENTER_SIZE • nms_threshold和confidence_threshold的范围为0.0~1.0 【量化工具支持】 是
39	Reorg	实时物体检测	【输入】 1个输入 【参数】 <ul style="list-style-type: none"> • stride: 可选, 类型: uint32, 默认为2 • reverse: 可选, 类型: bool, 默认为false 【约束】 只用于YOLOV2 【量化工具支持】 否
40	Reverse	逆转	【输入】 1个输入 【参数】 axis: 可选, 类型: int32, 默认为1; 控制需翻转的数据轴, 内容的布局不会被颠倒 【约束】 无限制 【量化工具支持】 否
41	LeakyRelu	LeakyRelu激活函数	【输入】 1个输入 【参数】 同Relu 【约束】 无限制 【量化工具支持】 是

序号	算子	含义	边界
42	YOLODetectionOutput	YOLO网络检测输出	<p>【输入】 4个输入</p> <p>【参数】</p> <ul style="list-style-type: none"> • num_classes: 必选, 类型: int32, 要预测的类数 • share_location: 可选, 类型: bool, 默认为true (表示不同类间共享bounding box) • background_label_id: 可选, 类型: int32, 默认为0 • nms_param: 可选, 非最大抑制 • save_output_param: 可选, 用于保存检测结果 • code_type: 可选, 默认为CENTER_SIZE • variance_encoded_in_target: 可选, 类型: bool, 默认为true; 如果为true, 方差编码在目标中, 否则需要相应地调整预测偏移量 • keep_top_k: 可选, 类型: int32, 在nms步骤后每个图像要保留的总bbox数 • confidence_threshold: 可选, 类型: float, 仅考虑置信度大于阈值的检测; 如果没有设置, 考虑所有的box • nms_threshold: 可选, 类型: float • top_k: 可选, 类型: int32 • boxes: 可选, 类型: int32, 默认为1 • relative: 可选, 类型: bool, 默认为true • objectness_threshold, 可选, 类型: float, 默认为0.5 • class_threshold: 可选, 类型: float, 默认为0.5 • biases: 数组 • general_nms_param: 可选 <p>【约束】</p> <ul style="list-style-type: none"> • 只用于YOLOV2 • classNUM<10240; anchorBox <= 8

序号	算子	含义	边界
			<ul style="list-style-type: none">• $W \leq 1536$• yolodetectionoutput的上层必须为yoloregion算子 【量化工具支持】 否

B Tensorflow 算子边界

“.om”模型支持的Tensorflow算子边界如[表B-1](#)所示。

表 B-1 TensorFlow 算子边界

序号	Python API	C++ API	边界
1	tf.nn.avg_pool	AvgPool Type: Mean	<p>【参数】</p> <ul style="list-style-type: none"> value: 4-D tensor, 格式: [batch, height, width, channels], 数据类型: float32 ksize: 包含四个int的列表或元组, 其中每个值对应相应维度的窗口大小 strides: 包含四个int的列表或元组, 其中每个值对应相应维度的滑动步长 padding: 类型: string, 值必须为'VALID' 或' SAME' data_format: 类型: string, 值为'NHWC' (默认值) 或'NCHW' name: 可选参数, 类型: string <p>【约束】</p> <ul style="list-style-type: none"> kernelH<=inputH+padTop+padBottom kernelW<=inputW+padLeft+padRight padTop<windowH padBottom<windowH padLeft<windowW padRight<windowW <p>除公共约束外, 还需满足下列某一场景支持范围:</p> <p>global pool模式只支持下列场景:</p> <ul style="list-style-type: none"> outputH==1 && outputW==1 && kernelH>=inputH && kernelW>=inputW inputH*inputW<=10000 <p>【输出】</p> <p>1个tensor, 数据类型与value相同</p> <p>【量化工具支持】</p> <p>是</p>
2	tf.nn.max_pool	MaxPool	同tf.nn.avg_pool

序号	Python API	C++ API	边界
3	tf.nn.conv2d	Conv2D	<p>【参数】</p> <ul style="list-style-type: none"> value: 4-D tensor, 格式: [batch, height, width, channels] 数据类型: float32 filter: 1个常量tensor, 数据类型与维度与value相同, [filter_height, filter_width, in_channels, out_channels] strides: 非空, 包含四个int的列表或元组, 其中每个值对应相应维度的滑动步长 padding: 非空, 类型: string, 值必须为'VALID'或'SAME' data_format: 非空, 类型: string, 值为'NHWC'(默认值)或'NCHW' dilations: 可选参数, 1个int列表(长度为4); 默认为[1, 1, 1, 1], 对应输入的每一维; 如果k>1, 相应维度做filter间跳过k-1个单元; 维度顺序由data_format决定; dilations的batch与depth维度上的值必须是1 name: 可选参数, 类型: string <p>【约束】</p> <ul style="list-style-type: none"> $(inputW + padWHead + padWTail) \geq ((FilterW - 1) * dilationW + 1)$ $(inputW + padWHead + padWTail) / StrideW + 1 \leq INT32_MAX$ $(inputH + padHHead + padHTail) \geq ((FilterH - 1) * dilationH + 1)$ $(inputH + padHHead + padHTail) / StrideH + 1 \leq INT32_MAX$ $0 \leq Pad < 256, 0 < FilterSize < 256, 0 < Stride < 64, 1 \leq dilationsize < 256$ <p>【输出】 Tensor, 数据类型与Value相同</p> <p>【量化工具支持】 是</p>

序号	Python API	C++ API	边界
4	tf.concat	Concat	<p>【参数】</p> <ul style="list-style-type: none"> values: 输入，包含tensor对象的列表或单个tensor，除要拼接的维度外，其他维度上的值要一致 axis: 0-D tensor，类型：int32，指定要拼接的维度，范围在[-rank (values) , rank (values)]；python中，索引以0开始，axis为正值时，表示对第axis维拼接；axis为负值时，对第axis+rank (values) 维拼接 <p>【约束】</p> <p>输入的tensor，除了进行concat的维度外，其他维度的size必须相等</p> <p>输入的的tensor个数范围属于[1, 1000]</p> <p>【输出】</p> <p>1个tensor，为输入tensors拼接后的结果</p> <p>【量化工具支持】</p> <p>是</p>

序号	Python API	C++ API	边界
5	tf.matmul	MatMul	<p>【参数】</p> <ul style="list-style-type: none"> • a: 非常量tensor, rank\geq2, 类型: float32 • b: 常量tensor, 类型与rank同a • transpose_a: 如果属性为True, a在乘法前做转置 • transpose_b: 如果属性为True, b在乘法前做转置; transpose_a属性为False, transpose_b属性也为False • adjoint_a: 如果属性为True, a在乘法前被共轭和转置 • adjoint_b: 如果属性为True, b在乘法前被共轭和转置 • a_is_sparse: 如果属性为True, a被看做是稀疏矩阵 • b_is_sparse: 如果属性为True, b被看做是稀疏矩阵 • name: 可选参数 <p>【约束】</p> <ul style="list-style-type: none"> • weights的转置属性为false • 不能支持两个tensor相乘, 只能支持1个tensor乘以1个常量 <p>【输出】</p> <p>1个tensor, 类型同a与b</p> <p>【量化工具支持】</p> <p>否</p>

序号	Python API	C++ API	边界
6	tf.nn.fused_batch_norm	FusedBatchNorm	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入, 4-D tensor, 类型: float32 • scale: 1-D tensor, 用于缩放 • offset: 1-D tensor, 偏差 • mean: 1-D tensor, 用于推理总体均值 • variance: 1-D tensor, 用于推理总体方差 • epsilon: 在x的方差中添加的1个小的浮点数 • data_format: x的数据格式, 值为 'NHWC' (默认) 或 'NCHW' • is_training: bool值, 用于指定操作是用于训练还是推断 • name: 操作的名称, 可选参数 <p>【约束】 scale, bias, mean, var的shape只支持 (1, C, 1, 1), 且c维度与input的c维度相等</p> <p>【输出】</p> <ul style="list-style-type: none"> • y: 标准化、缩放、偏移x的4-D tensor • batch_mean: 1-D tensor, 表示x的均值 • batch_var: 1-D tensor, 表示x的方差 <p>【量化工具支持】 否</p>
7	tf.abs	Abs	<p>【参数】</p> <ul style="list-style-type: none"> • x: tensor或稀疏tensor, 类型: float32 • name: 操作的名称, 可选参数 <p>【约束】 无限制</p> <p>【输出】 返回x的绝对值, tensor或稀疏tensor, 尺寸与类型同x</p> <p>【量化工具支持】 是</p>

序号	Python API	C++ API	边界
8	tf.image.resize_nearest_neighbor	ResizeNearestNeighbor	<p>【参数】</p> <ul style="list-style-type: none"> images: 4维tensor, [batch, height, width, channels]; 3维tensor, [height, width, channels]; 类型: float32 size: 1维tensor, 常量, 2个元素 (新的高宽) method: ResizeMethod.NEARESTNEIGHBOR align_corners: bool值, 默认为False; 如果为True, 输入和输出的4个角像素的中心对齐, 保留角像素处的值 <p>【约束】 无限制</p> <p>【输出】 shape同输入, 类型: float</p> <p>【量化工具支持】 否</p>
9	tf.image.resize_bilinear	ResizeBilinear	<p>【参数】</p> <ul style="list-style-type: none"> images: 4维非常量tensor, [batch, height, width, channels]; 类型: float32 size: 1维tensor, 常量, 2个元素 (新的高宽) method: ResizeMethod.BILINEAR align_corners: bool值, 默认为False; 如果为True, 输入和输出的4个角像素的中心对齐, 保留角像素处的值 <p>【约束】 $(outputH * outputW) / (inputH * inputW) > 1/7$</p> <p>【输出】 shape同输入, 类型: float</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
10	tf.cast	Cast	<p>【输入】</p> <p>类型: float32, int32, bool, int64, int16, int8, uint8, uint16, double</p> <p>【参数】</p> <ul style="list-style-type: none"> • x: tensor或sparseTensor或indexedSlices • dtype: 目标类型, 同x支持的数据类型 • name: 名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>tensor或sparseTensor或indexedSlices, 同输入的dtype、shape</p> <p>【量化工具支持】</p> <p>无需支持</p>
11	tf.nn.depthwise_conv2d	DepthwiseConv2dNative	<p>【参数】</p> <ul style="list-style-type: none"> • input: 4维 • filter: 4维, 常量, 数据格式: [filter_height, filter_width, in_channels, channel_multiplier] • strides: 输入的每个维度的滑动窗口的步长, 属性必须是list (int), 且大小为4 • padding: 类型: string, 值为'VALID' 或 'SAME' • rate: 1维, 大小为2; 等值卷积中在height和width维度上对输入值进行采样的扩张率; 如果值大于1, 则步长的所有值必须为1 • data_format: 输入的数据格式, 可以是NHWC (默认) 或NCHW • name: 名称 (可选) <p>【约束】</p> <p>filterN=inputC=group</p> <p>【输出】</p> <p>4维tensor, 形状与data_format一致, 例如, 对于NHWC格式, 形状是 [batch, out_height, out_width, in_channels * channel_multiplier]</p> <p>【量化工具支持】</p> <p>是</p>

序号	Python API	C++ API	边界
12	tf.reshape	Reshape	<p>【参数】</p> <ul style="list-style-type: none"> • tensor: 输入, 1个tensor • shape: 定义输出的shape, 常量tensor, 数据类型: int64, int • name: 此操作的名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>1个tensor, 类型同输入</p> <p>【量化工具支持】</p> <p>是</p>
13	tf.squeeze	Squeeze	<p>【参数】</p> <ul style="list-style-type: none"> • input: 1个tensor, 非常量输入 • axis: 1个int型列表, 指定要移除的维度, 默认为[]; 不能指定非1的维度 • name: 此操作的名称 (可选) • squeeze_dims: 不推荐使用的参数, axis和dim不能同时存在 <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>1个tensor, 与input的类型、数据相同, 但删除了1个或多个值为1的维度</p> <p>【量化工具支持】</p> <p>是</p>
14	tf.expand_dims	ExpandDims	<p>【参数】</p> <ul style="list-style-type: none"> • inuput: 输入, 1个tensor • axis: 0-D (标量), 指定扩展input形状的维度索引 • name: 输出tensor的名称 • dim: 0-D (标量), 相当于axis, 被弃用 <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>1个tensor, 与input数据相同, shape增加了一维 (值为1)</p> <p>【量化工具支持】</p> <p>是</p>

序号	Python API	C++ API	边界
15	tf.greater	Greater	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入, 1个tensor • y: 标量 • name: 此操作的名称 (可选) <p>【约束】</p> <p>支持广播broadcast, 对比x和y的shape, 在同一纬度上, dim[x]只能相同或者一方为1或者一方缺失</p> <p>【输出】</p> <p>1个tensor, 类型: bool</p> <p>【量化工具支持】</p> <p>无需支持</p>
16	tf.nn.relu	Relu	<p>【参数】</p> <ul style="list-style-type: none"> • features: 非常量输入, tensor • name: 此操作的名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>1个tensor, 类型同features</p> <p>【量化工具支持】</p> <p>是</p>
17	tf.nn.relu6	Relu6	<p>【参数】</p> <ul style="list-style-type: none"> • features: 非常量输入, tensor • name: 此操作的名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>1个tensor, 类型同features</p> <p>【量化工具支持】</p> <p>是</p>

序号	Python API	C++ API	边界
18	tf.nn.leaky_relu	/	<p>【参数】</p> <ul style="list-style-type: none"> features: 非常量输入, tensor, 表示预激活的值 alpha: $x < 0$时激活函数的斜率 name: 此操作的名称 (可选) <p>【约束】 无限制</p> <p>【输出】 激活值</p> <p>【量化工具支持】 是</p>
19	tf.exp	exp	<p>【参数】</p> <ul style="list-style-type: none"> x: 1个输入tensor, 类型float32, double name: 此操作的名称 (可选) <p>【约束】 无限制</p> <p>【输出】 1个tensor, 类型同x</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
20	tf.nn.conv2d_transpose	Conv2DBackpropInput	<p>【参数】</p> <ul style="list-style-type: none"> value: 输入, 4-D tensor, 数据格式: NHWC ([batch, height, width, in_channels]) 或 NCHW ([batch, in_channel, height, width]) filter: 4-D tensor, 常量, shape: [height, width, output_channels, in_channels] output_shape: 1-D tensor, 表示输出的shape strides: int型列表, 非空, 输入的每个维度的滑动窗口的步长 padding: 类型: string, 值为 'VALID' 或 'SAME', 非空 data_format: 类型: string, 'NHWC' 或 'NCHW', 非空 name: 输出名 (可选) <p>【约束】</p> <ul style="list-style-type: none"> group = 1 dilation = 1 filterH - padHHead - 1 >= 0 filterW - padWHead - 1 >= 0 <p>还有一条约束涉及中间变量, 公式如下:</p> <ol style="list-style-type: none"> $a = \text{ALIGN}(\text{filter_num}, 16) * \text{ALIGN}(\text{filter_c}, 16) * \text{filter_h} * \text{filter_w} * 2;$ 如果 $\text{ALIGN}(\text{filter_c}, 16) \% 32 = 0$, $a = a/2$ $\text{conv_input_width} = (\text{反卷积输入}w - 1) * \text{strideW} + 1$ $b = (\text{conv_input_width}) * \text{filter_h} * \text{ALIGN}(\text{filter_num}, 16) * 2 * 2$ $a + b \leq 1024 * 1024$ <p>【输出】 1个tensor, 类型同value</p> <p>【量化工具支持】 是</p>

序号	Python API	C++ API	边界
21	tf.sigmoid	Sigmoid	<p>【参数】</p> <ul style="list-style-type: none"> x: 1个tensor, 输入 name: 此操作的名称 (可选) <p>【约束】 无限制</p> <p>【输出】 1个tensor, 类型同value</p> <p>【量化工具支持】 是</p>
22	tf.add	Add	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入, 1个tensor, 类型: float32、int32 y: 输入, 1个tensor, 类型同x; 两个输入为常量时, 1个输入为标量 name: 此操作的名称 (可选) <p>【约束】 支持两组输入的维度不一致, 进行广播操作 (广播即维度补齐) 目前支持以下几种广播场景:</p> <ul style="list-style-type: none"> NHWC+NHWC, NHWC+scalar NHWC +1 1 1 1 NHWC+W和HWC+W和HW+W (备注, W维度做广播) NCHW + NH1C和HWC + H1C和HW + H1 HWC + 1 WC (备注, H维度做广播) <p>说明 两个tensor的输入顺序可以互换。</p> <p>【输出】 1个tensor, 类型同y</p> <p>【量化工具支持】 是</p>

序号	Python API	C++ API	边界
23	tf.multiply	Multiply Type: Mul	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入, 1个tensor, 类型: float32、int32 y: 输入, 1个tensor, 类型同x; 两个输入为常量时, 维度必须一致, 为标量或者1-D tensor name: 此操作的名称 (可选) <p>【约束】</p> <p>支持两组输入的维度不一致, 进行广播操作 (广播即维度补齐), 目前支持以下几种广播场景:</p> <ul style="list-style-type: none"> NHWC+NHWC, NHWC+scalar NHWC +1 1 1 1, NHWC+W和HWC+W和HW+W (备注, W维度做广播) NCHW + NH1C和HWC + H1C和HW + H1 HWC + 1 WC (备注, H维度做广播) <p>说明</p> <p>两个tensor的输入顺序可以互换。</p> <p>【输出】</p> <p>1个tensor</p> <p>【量化工具支持】</p> <p>是</p>

序号	Python API	C++ API	边界
24	tf.subtract	Subtract Type: Sub	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入, 1个tensor y: 输入, 1个tensor, 类型同x; 两个输入, 可支持常量或非常量 name: 此操作的名称 (可选) <p>【约束】</p> <p>支持两组输入的维度不一致, 进行广播操作 (广播即维度补齐), 目前支持以下几种广播场景:</p> <ul style="list-style-type: none"> NHWC+NHWC, NHWC+scalar NHWC +1 1 1 1 NHWC+W和HWC+W 和HW+W (备注, W维度做广播) NCHW + NH1C和HWC + H1C和HW + H1 HWC + 1 WC (备注, H维度做广播) <p>说明</p> <p>两个tensor的输入顺序可以互换。</p> <p>【输出】</p> <p>1个tensor</p> <p>【量化工具支持】</p> <p>是</p>
25	tf.nn.bias_add	BiasAdd	<p>【参数】</p> <ul style="list-style-type: none"> value: 输入, 1个tensor, 非常量 bias: 1-D tensor, 常量, 尺寸与value的最后一维一致; 除非value为量化类型, 否则类型需同value data_format: 类型: string, 'NHWC' 或 'NCHW' name: 此操作的名称 (可选) <p>【约束】</p> <ul style="list-style-type: none"> C < 10000 input和bias的数据排布要一致 当在c通道上加bias时, input和bias的C维度大小要一致 <p>【输出】</p> <p>1个tensor, 类型同value</p> <p>【量化工具支持】</p> <p>是</p>

序号	Python API	C++ API	边界
26	tf.nn.lrn	LRN	<p>【参数】</p> <ul style="list-style-type: none"> input: 输入, 4-D tensor, 类型: float32 depth_radius: 0-D int型, 默认值为5, 1-D标准化窗口的半宽 bias: 可选参数, float型, 默认为1; 偏移 (通常为正值, 以避免除以0) alpha: 可选参数, float型, 默认为1; 比例因子, 通常为正值 beta: 可选参数, float型, 默认为0.5; 1个指数 name: 此操作的名称 (可选) <p>【约束】</p> <ul style="list-style-type: none"> depth_radius > 0, 且必须为奇数 通道间: 当depth_radius ∈ [1, 15]时, alpha > 0.00001且beta > 0.01, 否则alpha和beta为任意值; 当C维度大于1776时, depth_radius < 1728 <p>【输出】 1个tensor, 类型同input</p> <p>【量化工具支持】 是</p>
27	tf.nn.elu	Elu	<p>【参数】</p> <ul style="list-style-type: none"> features: 输入tensor, 非常量 name: 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 输出tensor, 类型同features</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
28	tf.rsqrt	Rsqrt	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入tensor • name: 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 输出tensor, 类型同x</p> <p>【量化工具支持】 是</p>
29	tf.log	Log	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入tensor, 类型: float32 • name: 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 输出tensor, 类型同x</p> <p>【量化工具支持】 否</p>
30	tf.tanh	Tanh	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入tensor, 非常量 • name: 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 输出tensor, 类型同x</p> <p>【量化工具支持】 是</p>

序号	Python API	C++ API	边界
31	tf.slice	Slice	<p>【参数】</p> <ul style="list-style-type: none"> input_: 输入tensor begin: tensor, 类型: int32或int64 size: tensor, 类型: int32或int64 name: 名称 (可选) <p>【约束】</p> <p>输入tensor元素个数不超过INT32_MAX</p> <p>【输出】</p> <p>输出tensor, 类型同input_</p> <p>【量化工具支持】</p> <p>是</p>
32	tf.split	Split	<p>【参数】</p> <ul style="list-style-type: none"> value: 输入Tensor num_or_size_splits: 不支持这个参数 axis: 标准, 整型, 指定输出维度 name: string; 名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>List, 包含split后的各Tensor</p> <p>【量化工具支持】</p> <p>是</p>
33	tf.nn.softplus	Softplus	<p>【参数】</p> <ul style="list-style-type: none"> features: 输入Tensor; 数据类型: `float32` name: string; 名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>输出Tensor, 数据类型与features相同</p> <p>【量化工具支持】</p> <p>否</p>

序号	Python API	C++ API	边界
34	tf.nn.softsign	Softsign	<p>【参数】</p> <ul style="list-style-type: none"> features: 输入Tensor; 数据类型: `float32` name: string; 名称(可选) <p>【约束】 无限制</p> <p>【输出】 输出Tensor, 数据类型与features相同</p> <p>【量化工具支持】 否</p>
35	tf.pad	Pad/ MirrorPad/ PadV2	<p>【参数】</p> <ul style="list-style-type: none"> tensor: 4-D Tensor; 数据类型: `float32`, `int32` paddings: Tensor, 常量, 数据类型: `int32` mode: string, 值为"CONSTANT", 或"REFLECT", 或"SYMMETRIC" name: string; 名称(可选) constant_values: Pad默认填充的值, 标量, 数据类型与tensor相同 <p>【约束】 CONSTANT模式时, $0 \leq \text{PAD} \leq 128$, $0 < W \leq 3000$</p> <p>【输出】 输出Tensor, 数据类型与tensor相同</p> <p>【量化工具支持】 是</p>

序号	Python API	C++ API	边界
36	tf.fake_quant_with_min_max_vars	FakeQuantWithMinMaxVars	<p>【参数】</p> <ul style="list-style-type: none"> inputs: 输入Tensor; 数据类型: `float32` min: Tensor, 数据类型: `float32` max: Tensor, 数据类型: `float32` num_bits: 标量, 整型, 默认值`8` narrow_range: bool (可选), 默认值`False` name: string; 名称 (可选) <p>【约束】 -65504<=min<=65504, -65504<=max<=65504</p> <p>【输出】 输出Tensor, 数据类型: `float32`</p> <p>【量化工具支持】 否</p>
37	tf.reduce_max	Max	<p>【参数】</p> <ul style="list-style-type: none"> input_tensor: 输入tensor; 数据类型: `float32`, `int64`, `int32`, `uint8`, `uint16`, `int16`, `int8` axis: 1-D list或标量, 数据类型整型 keepdims: bool, 是否保留维度为1 name: string; 名称 (可选) reduction_indices: axis旧的别名 (不推荐) keep_dims: keepdims别名 (不推荐) <p>【约束】</p> <ul style="list-style-type: none"> 当输入的tensor维数等于4时: 输入axis={3, {1, 2, 3}}, keepDims=true, $H*W*16*2 \leq 16*1024$ 当输入的tensor维数等于2时: 输入axis={1, {1}}, keepDims=true, $H*W*CEIL(C, 16)*16*2 \leq 16*1024$ <p>【输出】 输出reduce后的Tensor, 数据类型同input_tensor</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
38	tf.strided_slice	StridedSlice	<p>【参数】</p> <ul style="list-style-type: none"> input_: 1个Tensor begin: 1-D Tensor, 数据类型: `int32` end: 1-D Tensor, 数据类型: `int32` strides: 1-D Tensor, 数据类型: `int32` begin_mask: 标量, 数据类型: `int32` end_mask: 标量, 数据类型: `int32` ellipsis_mask: 标量, 数据类型: `int32` new_axis_mask: : 标量, 数据类型: `int32` shrink_axis_mask: 标量, 数据类型: `int32` var: 与input_或None对应的变量 name: string; 名称 (可选) <p>【约束】 strides不允许为0</p> <p>【输出】 输出Tensor, 数据类型同input_</p> <p>【量化工具支持】 否</p>
39	tf.reverse	Reverse	<p>【参数】</p> <ul style="list-style-type: none"> tensor: 1个包含Tensor的列表, axis: Tensor reverse轴向, 数据类型: `int32`, `int64` name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 1个Tensor, 数据类型同tensor</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
40	tf.realdiv	RealDiv	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入Tensor; 数据类型: float32` • y: 输入Tensor; 数据类型: float32` • name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 输出Tensor, 数据类型同x</p> <p>【量化工具支持】 是</p>
41	tf.stack	Stack	<p>【参数】</p> <ul style="list-style-type: none"> • values: 包含tensor对象 (形状与类型相同, 类型: float32, int32) 的列表 • axis: 整数, 沿axis维度堆叠, 默认是第一维, 必须设置 • name: 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 堆叠后的tensor, 类型同values 属性T、N和axis必须存在</p> <p>【量化工具支持】 否</p>
42	tf.unstack	Unpack	<p>【参数】</p> <ul style="list-style-type: none"> • value: 输入tensor, 值>0, 类型: float32、int32 • num: 整数, 表示axis维度的长度, 默认为None • axis: 整数, 沿axis拆分, 默认是第一维 • name: 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 从value中拆分出的包含tensor对象的列表</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
43	tf.transpose	Transpose	<p>【参数】</p> <ul style="list-style-type: none"> • a: 输入tensor • perm: a的维数的排列 • name: 名称 (可选) • conjugate: 可选, 类型: bool, 默认为False; 设置为True时, 等同于tf.conj (tf.transpose (input)) <p>【约束】 无限制</p> <p>【输出】 被转置后的tensor</p> <p>【量化工具支持】 是</p>
44	tf.space_to_batch_nd	SpaceToBatchND	<p>【参数】</p> <ul style="list-style-type: none"> • input: 1个Tensor, 是N维的并且具有形状input_shape = [batch] + spatial_shape + remaining_shape, 其中spatial_shape有M维度; 支持数据类型为: uint8, int8, int16, uint16, int32, int64, float32 • block_shape: 1个Tensor, 必须是以下类型之一: int32, int64; 1-D, shape为[M], 所有值必须>=1 • paddings: 1个Tensor, 必须是以下类型之一: int32, int64; 二维, shape为[M, 2], 所有值必须>=0 <p>【约束】 当tensor维数为4时: blockShape的长度必须等于2, paddings的长度必须等于4</p> <ul style="list-style-type: none"> • blockShape元素的大小必须要大于等于1, paddings元素值的大小必须大于等于0 • padding后的h维度要能够被blockShape[0]整除, padding后的w维度要能够被blockShape[1]整除 <p>【输出】 1个Tensor, 与input具有相同的类型</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
45	tf.batch_to_space_nd	BatchToSpaceND	<p>【参数】</p> <ul style="list-style-type: none"> input: 1个Tensor, 是N维的并且具有形状input_shape = [batch] + spatial_shape + remaining_shape, 其中spatial_shape有M维度; 支持数据类型为: uint8, int8, int16, uint16, int32, int64, float32 block_shape: 1个Tensor; 必须是以下类型之一: int32, int64; 1-D, shape为[M], 所有值必须>=1 crops: 1个Tensor; 必须是以下类型之一: int32, int64; 二维, shape为[M, 2], 所有值必须>=0 <p>【约束】</p> <ul style="list-style-type: none"> blockShape和crops的元素值数据类型必须是int32, 当tensor维数为4时: blockShape的长度必须等于2, crops的长度必须等于4 blockShape元素的大小必须要大于等于1, crops元素值的大小必须大于等于0, crops数组的大小必须满足 $\text{crop_start}[i] + \text{crop_end}[i] < \text{block_shape}[i] * \text{input_shape}[i+1]$ <p>【输出】 1个Tensor, 与images具有相同的类型</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
46	tf.extract_image_patches	ExtractImagePatches	<p>【参数】</p> <ul style="list-style-type: none"> images: 1个Tensor, 支持数据类型: float32, int32, int64, uint8, int8, uint16, int16; 4-D Tensor shape: [batch, in_rows, in_cols, depth] ksize: 1个整型list, 长度>=4 strides: 1个整型list, 必须是: [1, stride_rows, stride_cols, 1] rate: 1个整型list, 必须是: [1, rate_rows, rate_cols, 1] padding: string, 取值: "VALID" 或者 "SAME", "VALID" 表示所取的patch区域必须完全包含在原始图中"SAME"表示取超出原始图像的部分, 以0填充该部分 name: 名称 (可选) <p>【约束】 无约束</p> <p>【输出】 1个Tensor, 与images具有相同的类型</p> <p>【量化工具支持】 否</p>
47	tf.floormod	FloorMod	<p>【参数】</p> <ul style="list-style-type: none"> x: 1个Tensor, 支持数据类型: float32, int32 y: 1个Tensor, 与x具有相同类型 name: 名称 (可选) <p>【约束】 由于支持广播broadcast, 对比x和y的shape, 在同一纬度上, dim[x]只能相同或者一方为1或者一方缺失</p> <p>【输出】 1个Tensor, 与x具有相同的类型</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
48	tf.nn.softmax	Softmax	<p>【参数】</p> <ul style="list-style-type: none"> logits: 1个非空的Tensor, 支持数据类型: float32 axis: 在其上执行维度softmax。默认值为-1, 表示最后1个维度, 不超过logits维度 name: 名称 (可选) dim: axis的已弃用的别名 <p>【约束】</p> <ul style="list-style-type: none"> 输入4维时可以针对每一维做softmax: 根据分类的轴不同, 计算边界分别为: axis=1时, $c \leq ((256 \times 1024 / 4) - 8 \times 1024 - 256) / 2$ axis=0时, $n \leq (56 \times 1024 - 256) / 2$ axis=2时, $W=1, 0 < h < (1024 \times 1024 / 32)$ axis=3时, $0 < W < (1024 \times 1024 / 32)$ 输入维度不足4维时, 仅支持对最后一维做softmax计算, 并且最后一维不超过46080 <p>【输出】 1个Tensor, 与logits具有相同的类型和shape</p> <p>【量化工具支持】 是</p>
49	tf.math.pow	Power	<p>【参数】</p> <ul style="list-style-type: none"> x: 1个Tensor, 支持数据类型: float32 y: 1个Tensor, 支持数据类型: float32 name: 名称 (可选) <p>【约束】 power!=1 scale*x+shift>0</p> <p>【输出】 1个Tensor</p> <p>【量化工具支持】 是</p>

序号	Python API	C++ API	边界
50	tf.placeholder	-	<p>【参数】</p> <ul style="list-style-type: none"> dype: 数据类型（必须） shape: Tensor的维度和大小（必须） <p>【约束】 无限制</p> <p>【输出】 1个Tensor</p> <p>【量化工具支持】 否</p>
51	tf.shape	Shape	<p>【参数】</p> <ul style="list-style-type: none"> input: 1个Tensor或`SparseTensor` name: string; 名称（可选） out_type输出Tensor数据类型, `int32`或`int64`（可选, 默认为`int32`） <p>【约束】 无限制</p> <p>【输出】 1个Tensor, 输出数据类型为out_type</p> <p>【量化工具支持】 无需支持</p>
52	tf.math.argmax x	ArgMax	<p>【参数】</p> <ul style="list-style-type: none"> input: 1个Tensor, 支持数据类型: `int8`, `uint8`, `int16`, `uint16`, `int32`, `int64`, `float32` axis: 1个Tensor, 数据类型`int32`, `int64` out_type输出Tensor数据类型, `int32`或`int64`（可选, 默认为`int64`） name: string; 名称（可选） <p>【约束】 无限制</p> <p>【输出】 1个Tensor, 输出数据类型为out_type</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
53	tf.gather	Gather GatherV2	<p>【参数】</p> <ul style="list-style-type: none"> params: 1个Tensor, 维度必须大于`axis + 1` indices: 1个Tensor, 数据类型`int32`, `int64`, 范围[0, params.shape[axis]) axis: 输出Tensor数据类型, `int32`或`int64`, 指定indices选取的维度, rank=0 name: string, 名称(可选) <p>【约束】 无限制</p> <p>【输出】 1个Tensor, 输出数据类型与params相同</p> <p>【量化工具支持】 否</p>
54	tf.gather_nd	GatherNd	<p>【参数】</p> <ul style="list-style-type: none"> params: 1个Tensor, 维度必须大于`axis + 1` indices: 1个Tensor, 数据类型`int32`, `int64` name: string; 名称(可选) <p>【约束】</p> <ul style="list-style-type: none"> indices最后一维的大小不能超过params的维数 indices最后一维中的元素对应着params中的1个维度上的坐标, 必须满足坐标规则 indices中对应维度上的坐标不能超过维度的大小 <p>【输出】 1个Tensor, 输出数据类型与params相同</p> <p>【量化工具支持】 是</p>

序号	Python API	C++ API	边界
55	tf.math.floordiv	FloorDiv	<p>【参数】</p> <ul style="list-style-type: none"> x: 1个Tensor, 数据类型: `float32`, `int32`; y: 1个Tensor, 分母, 数据类型: `float32`, `int32`; name: string; 名称 (可选) <p>【约束】 由于支持广播broadcast, 对比x和y的shape, 在同一纬度上, dim[x]只能相同或者一方为1或者一方缺失</p> <p>【输出】 1个Tensor, floor (x/y)</p> <p>【量化工具支持】 否</p>
56	tf.range	Range	<p>【参数】</p> <ul style="list-style-type: none"> start: 开始标量, 数据类型 `float32`, `int32`, 必须为常量 limit: 结束标量, 数据类型 `float32`, `int32`, 必须为常量 delta: 步长标量, 数据类型 `float32`, `int32`, 必须为常量 dtype: 返回Tensor的数据类型 name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 1个1-D Tensor</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
57	tf.tile	Tile	<p>【参数】</p> <ul style="list-style-type: none"> input: 输入Tensor, 数据类型: `int8`, `uint8`, `int16`, `uint16`, `int32`, `int64`, `float32` multiples: 1-D Tensor, 长度须和input的秩相同, 数据类型: `int32`, 必须为常量 name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 1个Tensor</p> <p>【量化工具支持】 是</p>
58	tf.size	Size	<p>【参数】</p> <ul style="list-style-type: none"> input: 输入Tensor, 数据类型: float32` name: string; 名称 (可选) out_type: 输出Tensor数据类型, 默认`int32` <p>【约束】 无限制</p> <p>【输出】 1个Tensor, 类型由out_type指定</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
59	tf.fill	Fill	<p>【参数】</p> <ul style="list-style-type: none"> • dims: 1-D Tensor, 数据类型: `int32`, • value: 变量, 数据类型: `int32`, `float32` • name: string; 名称 (可选) <p>【约束】</p> <p>支持Constant、GivenTensor、Range、Diagonal、Gaussian、MSRA、Uniform、UniformInt、UniqueUniform、XavierFill这些填充模式, 在Uniform填充、UniformInt填充、UniqueUniform填充、xavier填充时, 生成的数值区间最大范围介于[min, max) 之间</p> <p>【输出】</p> <p>1个Tensor, 类型同value数据类型</p> <p>【量化工具支持】</p> <p>否</p>
60	tf.concat	Concat	<p>【参数】</p> <ul style="list-style-type: none"> • value: 1个包含Tensor的列表, 数据类型: `int32`, `float32` • axis: Tensor拼接轴向, 数据类型: `int32` • name: string; 名称 (可选) <p>【约束】</p> <p>输入的tensor, 除了进行concat的维度外, 其他维度的size必须相等</p> <p>输入的的tensor个数范围属于[1, 1000]</p> <p>【输出】</p> <p>1个Tensor</p> <p>【量化工具支持】</p> <p>是</p>

序号	Python API	C++ API	边界
61	tf.reverse	Reverse	<p>【参数】</p> <ul style="list-style-type: none"> • tensor: 1个包含Tensor的列表, • axis: Tensor reverse轴向, 数据类型: `int32` • name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 1个Tensor, 数据类型同tensor</p> <p>【量化工具支持】 否</p>
62	tf.reduce_sum	sum	<p>【参数】</p> <ul style="list-style-type: none"> • input_tensor: 输入Tensor, • axis: Tensor sum轴向, 数据类型: `int32` • keepdims: bool值, 是否保留维度 • name: string; 名称 (可选) • reduction_indices: axis值的string; 名称 • keep_dims: 不推荐使用, 参数 keepdims别名 <p>【约束】 无约束</p> <p>【输出】 输出Tensor, 数据类型同tensor</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
63	tf.math.maximum	Maximum	<p>【参数】</p> <ul style="list-style-type: none"> x: Tensor数据类型: `int32`, `int64`, `float32` y: Tensor数据类型同x name: string; 名称 (可选) <p>【约束】 由于支持广播broadcast, 对比x和y的shape, 在同一纬度上, dim[x]只能相同或者一方为1或者一方缺失</p> <p>【输出】 输出Tensor, 返回值 (x > y ? x : y) 数据类型同x</p> <p>【量化工具支持】 否</p>
64	tf.math.minimum	Minimum	<p>【参数】</p> <ul style="list-style-type: none"> x: Tensor数据类型: `int32`, `int64`, `float32` y: Tensor数据类型同x name: 名称 (可选) <p>【约束】 广播场景只支持下面两种: NHWC+scaler, NHWC+NHWC</p> <p>【输出】 输出Tensor, 返回值 (x < y ? x : y) 数据类型同x</p> <p>【量化工具支持】 否</p>
65	tf.clip_by_value	ClipByValue	<p>【参数】</p> <ul style="list-style-type: none"> t: Tensor clip_value_min: clip最小值 clip_value_max: clip最大值 name: string; 名称 (可选) <p>【约束】 min值要小于或者等于max值</p> <p>【输出】 输出Tensor, 返回值范围 【clip_value_min, clip_value_max】</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
66	tf.math.logical_not	LogicalNot	<p>【参数】</p> <ul style="list-style-type: none"> x: Tensor数据类型bool name: string; 名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>输出Tensor, 数据类型bool</p> <p>【量化工具支持】</p> <p>否</p>
67	tf.math.logical_and	LogicalAnd	<p>【参数】</p> <ul style="list-style-type: none"> x: Tensor数据类型bool y: Tensor数据类型bool name: string; 名称 (可选) <p>【约束】</p> <p>BroadCast只支持如下几种维度的广播, NHWC和[1, 1, 1, 1], [N, H, W, C], [N, H, W, 1], [1, H, W, C], [N, 1, 1, C]</p> <p>【输出】</p> <p>输出Tensor, 数据类型bool</p> <p>【量化工具支持】</p> <p>否</p>
68	tf.equal	Equal	<p>【参数】</p> <ul style="list-style-type: none"> x: Tensor y: Tensor, 数据类型同x name: string, 名称 (可选) <p>【约束】</p> <p>由于支持广播broadcast, 对比x和y的shape, 在同一纬度上右对齐的情况下, xdim[i]和ydim[i]只能相同或者一方为1或者一方缺失</p> <p>【输出】</p> <p>输出Tensor, 数据类型bool</p> <p>【量化工具支持】</p> <p>否</p>

序号	Python API	C++ API	边界
69	tf.square	Square	<p>【参数】</p> <ul style="list-style-type: none"> x: Tensor name: string, 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 输出Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>
70	tf.image.crop_and_resize	CropAndResize	<p>【参数】</p> <ul style="list-style-type: none"> image: 4-D Tensor; 数据类型: `float32`, `int8`, `int32`, `int64`, shape: `[num_boxes, 4]` boxes: 2-D Tensor; 数据类型: `float32`, shape: `[num_boxes]` box_ind: 1-D Tensor, 数据类型: `int32` crop_size: 1-D Tensor, 包含2个元素, 数据类型: `int32` method: string, 表示插值方法, 值为"bilinear" (默认), "nearest" name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 Tensor, 数据类型: `float32`</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
71	tf.math.top_k	TopKV2	<p>【参数】</p> <ul style="list-style-type: none"> input: Tensor, ≥ 1-D, 最后1个维度大小必须大于k, 数据类型: <code>`float32`</code>; k: 标量, ≥ 1; 数据类型: <code>`int32`</code> sorted: bool name: string; 名称 (可选) <p>【约束】 K一定要以常量传入</p> <p>【输出】</p> <ul style="list-style-type: none"> values: Tensor返回最后维度上的K个最大向量 indices: Tensor, values在input中的索引位置 <p>【量化工具支持】 否</p>
72	tf.invert_permutation	InvertPermutation	<p>【参数】</p> <ul style="list-style-type: none"> x: 1-D Tensor, 数据类型: <code>`int32`</code>, <code>`int64`</code>, name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
73	tf.multinomial	Multinomial	<p>【参数】</p> <ul style="list-style-type: none"> logits: 2-D Tensor, shape `[batch_size, num_classes]` num_samples: 标量, 抽样个数 seed: 随机数种子, 数据类型: `int32`, `int64`, name: string; 名称 (可选) output_dtype: 输出Tensor数据类型: 整型默认`int64` <p>【约束】 seed为0时产生随机数是动态的 输出数据行数等于输出数据的行, 输出数据的列数等num_samples</p> <p>【输出】 Tensor, 数据类型同output_dtype</p> <p>【量化工具支持】 否</p>
74	tf.reverse_sequence	ReverseSequence	<p>【参数】</p> <ul style="list-style-type: none"> input: 输入Tensor seq_lengths: 1-D Tensor; 数据类型: `int32`, `int64` seq_axis: 标量, , 数据类型: 整型 batch_axis: 标量 (可选), 数据类型: 整型 name: string; 名称 (可选) <p>【约束】</p> <ul style="list-style-type: none"> seq_lengths的长度必须等于input在batchAxis的元素数 seq_lengths的最大元素必须要小于等于seq_dim的元素数 seqAxis、batchAxis、seqDim、batchDim必须是int64类型 seqAxis与seqDim不可同时指定, batchAxis与batchDim不可同时指定 batchAxis和batchDim是可选参数, 缺省值为0, 权值个数需要为1 <p>【输出】 Tensor, 数据类型同input</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
75	tf.math.reciprocal	Reciprocal	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型`float32` name: string; 名称 (可选) <p>【约束】</p> <p>不支持输入数据中包含0</p> <p>【输出】</p> <p>Tensor, 数据类型同x</p> <p>【量化工具支持】</p> <p>否</p>
76	tf.nn.selu	Selu	<p>【参数】</p> <ul style="list-style-type: none"> features: 输入Tensor数据类型`float32`, name: string; 名称 (可选) <p>【约束】</p> <p>无约束</p> <p>【输出】</p> <p>Tensor, 数据类型同features</p> <p>【量化工具支持】</p> <p>否</p>
77	tf.math.acosh	Acosh	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型`float32` name: string; 名称 (可选) <p>【约束】</p> <p>无约束</p> <p>【输出】</p> <p>Tensor, 数据类型同x</p> <p>【量化工具支持】</p> <p>否</p>
78	tf.math.asinh	Asinh	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型`float32` name: string; 名称 (可选) <p>【约束】</p> <p>无约束</p> <p>【输出】</p> <p>Tensor, 数据类型同x</p> <p>【量化工具支持】</p> <p>否</p>

序号	Python API	C++ API	边界
79	tf.math.reduce_prod	Prod	<p>【参数】</p> <ul style="list-style-type: none"> input_tensor: 输入Tensor axis: reduce维度 keepdims: bool是否需要保存reduce维度 name: string; 名称 (可选) reduction_indices: 参数axis旧的别名 (不推荐) keep_dims: 参数keepdims别名 (不推荐) <p>【约束】 无约束</p> <p>【输出】 Tensor, reduce后的Tensor</p> <p>【量化工具支持】 否</p>
80	tf.math.sqrt	Sqrt	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor; 数据类型: `float32` name: string; 名称 (可选) <p>【约束】 无约束</p> <p>【输出】 Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
81	tf.math.reduce_all	All	<p>【参数】</p> <ul style="list-style-type: none"> input_tensor: 输入Tensor; 数据类型: bool axis: 指定reduce的维度 keepdims: bool值 name: string; 名称 (可选) reduction_indices: axis旧的别名 (不推荐) keep_dims: keepdims别名 (不推荐) <p>【约束】 无约束</p> <p>【输出】 Tensor, 数据类型同input_tensor</p> <p>【量化工具支持】 否</p>
82	tf.nn.l2_normalize	L2Normalize	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor; 数据类型: boolean axis: 指定normalize的维度轴向 <ul style="list-style-type: none"> 如果format为NCHW, 则axis必须为1 如果format为NHWC, 则axis必须为3 epsilon: 规范化的下限值.如果 $norm < \sqrt{\epsilon}$, 将使用 $\sqrt{\epsilon}$ 作为除数 name: string; 名称 (可选) dim: axis旧的别名 (不推荐) <p>【约束】 $H*W*2 < 256*1024/4$</p> <p>【输出】 Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
83	tf.keras.backend.hard_sigmoid	Hardsigmoid	<p>【参数】 x: 输入Tensor</p> <p>【约束】 无限制</p> <p>【输出】 输出Tensor, 返回值: `0.` if `x < -2.5`, `1.` if `x > 2.5` 当`-2.5 <= x <= 2.5`, 返回`0.2 * x + 0.5`</p> <p>【量化工具支持】 否</p>
84	tf.keras.layers.ThresholdedReLU	ThresholdedReLU	<p>【参数】 theta: 标量≥ 0, 数据类型: ; `float32`</p> <p>【约束】 无限制</p> <p>【输出】 Tensor</p> <p>【量化工具支持】 否</p>
85	tf.math.acos	Acos	<p>【参数】 x: 输入Tensor数据类型`float32`, `int32`, `int64` name: string; 名称 (可选)</p> <p>【约束】 输入数据范围 ($-1 \leq x \leq 1$), 输出数据 范围 ($0 \leq y \leq \pi$)</p> <p>【输出】 Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
86	tf.math.atan	Arctan	<p>【参数】 x: 输入Tensor数据类型`float32`, `int32`, `int64` name: string; 名称 (可选)</p> <p>【约束】 输入数据范围 ($-65504 \leq x \leq 65504$), 输出数据范围 ($-\pi/2 < y < \pi/2$)</p> <p>【输出】 Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>
87	tf.math.asin	Asin	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型`float32`, `int32`, `int64` name: string; 名称 (可选) <p>【约束】 输入数据范围 ($-1 \leq x \leq 1$), 输出数据范围 ($-\pi/2 \leq y \leq \pi/2$)</p> <p>【输出】 Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>
88	tf.math.atanh	Atanh	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型`float32`, `int32`, `int64` name: string; 名称 (可选) <p>【约束】 输入数据范围: $x \in (-1, 1)$</p> <p>【输出】 Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
89	tf.math.tan	Tan	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型`float32`, `int32`, `int64` name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 Tensor, 数据类型同x</p> <p>【量化工具支持】 否</p>
90	tf.math.logical_or	LogicalOr	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型: `bool` y: 输入Tensor数据类型: `bool` name: string; 名称 (可选) <p>【约束】 由于支持广播broadcast, 对比x和y的shape, 在同一纬度上右对齐的情况下, xdim[i]和ydim[i]只能相同或者一方为1或者一方缺失</p> <p>【输出】 Tensor, 数据类型: `bool`</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
91	tf.math.reduce_min	ReduceMin	<p>【参数】</p> <ul style="list-style-type: none"> input_tensor: 输入Tensor数据类型: `float32`, `int64`, `int32`, `uint8`, `uint16`, `int8`, `int16`, axis: reduce的维度轴向 keepdims: 标量, bool类型 name: string; 名称 (可选) reduction_indices: axis旧的别名 keep_dims: keepdims别名 (不推荐) <p>【约束】</p> <ul style="list-style-type: none"> 当输入的tensor维数等于4时: 输入 axis={3, {1, 2, 3}}, keepDims=true, $H*W*16*2 \leq 16*1024$ 当输入的tensor维数等于2时, 输入 axis={1, {1}}, keepDims=true, $H*W*CEIL(C, 16)*16*2 \leq 16*1024$ <p>【输出】 Tensor, 数据类型同input_tensor</p> <p>【量化工具支持】 否</p>
92	tf.math.negative	Neg	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型: `float32`, `int64`, `int32` name: string; 名称 (可选) <p>【约束】 输入数据范围 ($-65504 \leq x \leq 65504$), 输出数据范围 ($-65504 \leq y \leq 65504$)</p> <p>【输出】 Tensor, 输出=-x</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
93	tf.math.greater_equal	GreaterEqual	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor数据类型: `float32`, `int64`, `int32`, `uint8`, `uint16`, `int8`, `int16` y: 输入Tensor数据类型同x name: string; 名称 (可选) <p>【约束】 输入数据范围 (-65504<=x<=65504)</p> <p>【输出】 Tensor, 输出类型bool</p> <p>【量化工具支持】 否</p>
94	tf.space_to_depth	SpaceToDepth	<p>【参数】</p> <ul style="list-style-type: none"> input: 输入Tensor数据类型: `float32`, `int64`, `int32`, `uint8`, `int8` block_size: 标量, 整型, 值>=2 data_format: 数据类型: string值: "NHWC", "NCHW", "NCHW_VECT_C"; 默认值"NHWC" name: string; 名称 (可选) <p>【约束】 blockSize的大小必须大于等于1, 且能被H和W整除</p> <p>【输出】 Tensor, 输出类型input</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
95	tf.depth_to_space	DepthToSpace	<p>【参数】</p> <ul style="list-style-type: none"> input: 输入Tensor数据类型: `float32`, `int64`, `int32`, `uint8`, `int8` block_size: 标量, 整型, 值≥ 2 data_format: 数据类型: string值: "NHWC", "NCHW", "NCHW_VECT_C"; 默认值是 "NHWC" name: string; 名称 (可选) <p>【约束】</p> <p>blockSize必须大于等于1, 且 blockSize*blockSize必须能被C整除</p> <p>【输出】</p> <p>Tensor, 输出类型input</p> <p>【量化工具支持】</p> <p>否</p>
96	tf.math.round	Round	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor; 数据类型: `float32`, `int64`, `int32` name: string; 名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>Tensor, 输出类型同x, 输出shape同x</p> <p>【量化工具支持】</p> <p>否</p>
97	tf.math rint	Rint	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor; 数据类型: `float32`, `int64`, `int32`, `uint8`, `int8` name: string; 名称 (可选) <p>【约束】</p> <p>无限制</p> <p>【输出】</p> <p>Tensor, 输出类型同x, 输出shape同x</p> <p>【量化工具支持】</p> <p>否</p>

序号	Python API	C++ API	边界
98	tf.math.less	Less	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入Tensor数据类型: `float32`, `int64`, `int32`, `uint8`, `uint16`, `int8`, `int16` • y: 输入Tensor数据类型同x, • name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 Tensor, 输出类型bool</p> <p>【量化工具支持】 否</p>
99	tf.math.sinh	Sinh	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入Tensor数据类型: `float32` • name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 Tensor, 输出类型同x</p> <p>【量化工具支持】 否</p>
100	tf.math.cosh	Cosh	<p>【参数】</p> <ul style="list-style-type: none"> • x: 输入Tensor数据类型: `float32` • name: string; 名称 (可选) <p>【约束】 无限制</p> <p>【输出】 Tensor, 输出类型同x</p> <p>【量化工具支持】 否</p>

序号	Python API	C++ API	边界
101	tf.math.squared_difference	Squared_difference	<p>【参数】</p> <ul style="list-style-type: none"> x: 输入Tensor; 数据类型: `float32`, `int64`, `int32` y: 输入Tensor数据类型同x name: string; 名称 (可选) <p>【约束】</p> <p>广播模式只支持下列场景: 第1个的tensor_format是NCHW, 另1个的dim{}可以是[1, 1, 1, 1], [N, C, H, W], [N, 1, H, W], [1, C, H, W], [N, C, 1, 1], [1, C, 1, 1], [1, 1, H, W], [N, 1, 1, 1]这几种情况</p> <p>【输出】</p> <p>Tensor, 数据类型同x</p> <p>【量化工具支持】</p> <p>否</p>