

A child wearing a pilot's cap and goggles sits on the shoulder of a large, white, humanoid robot. The child is pointing their finger towards a large, glowing globe in the background. The globe features a world map overlay. The scene is set against a blue sky with streaks of light, suggesting a futuristic or space-themed environment.

昇腾CANN AI 应用开发实践

目录

TABLE OF CONTENTS

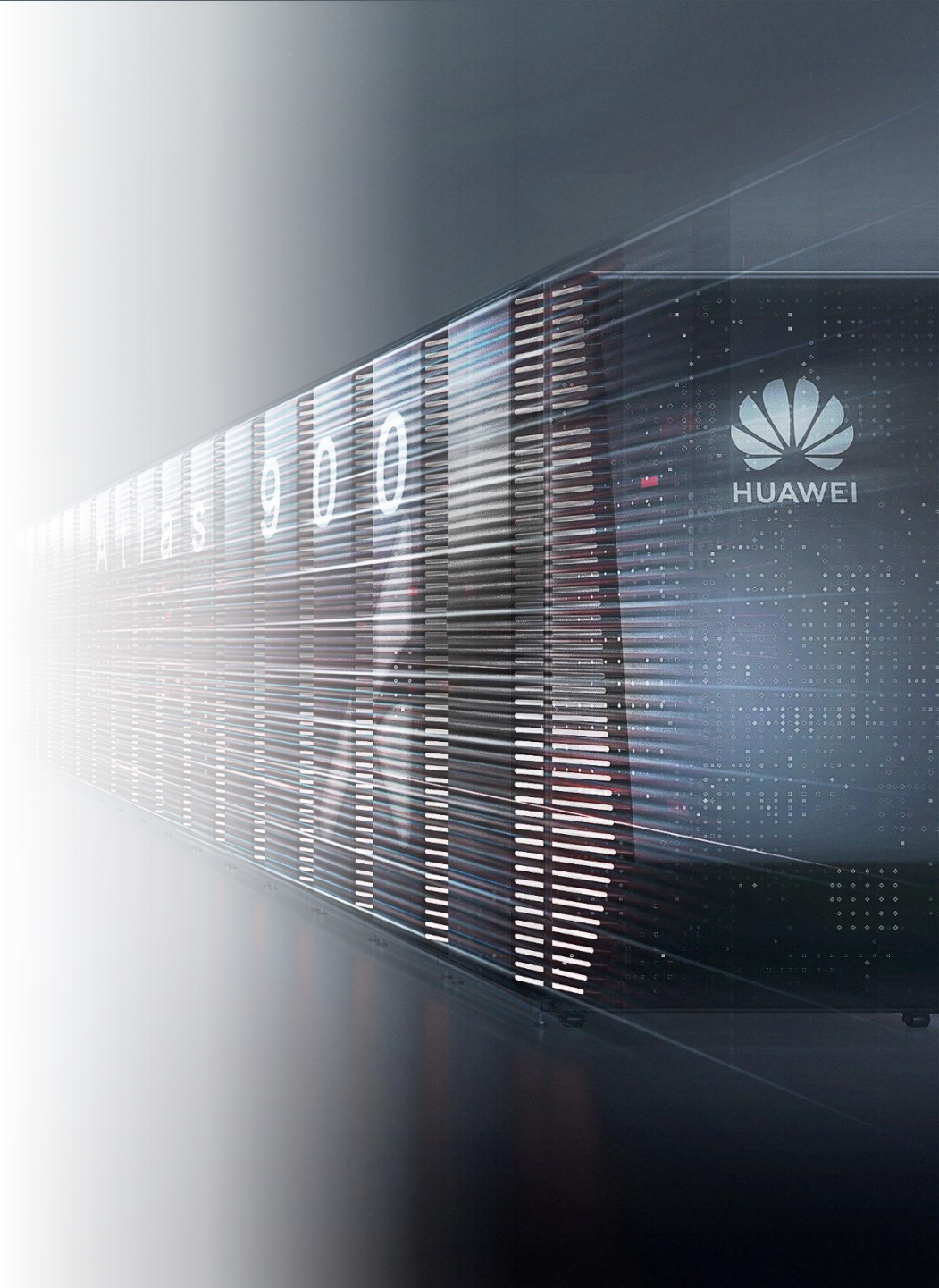
1 异构计算架构 CANN

2 CANN AI 应用开发流程

3 实战案例解析

4 性能&精度

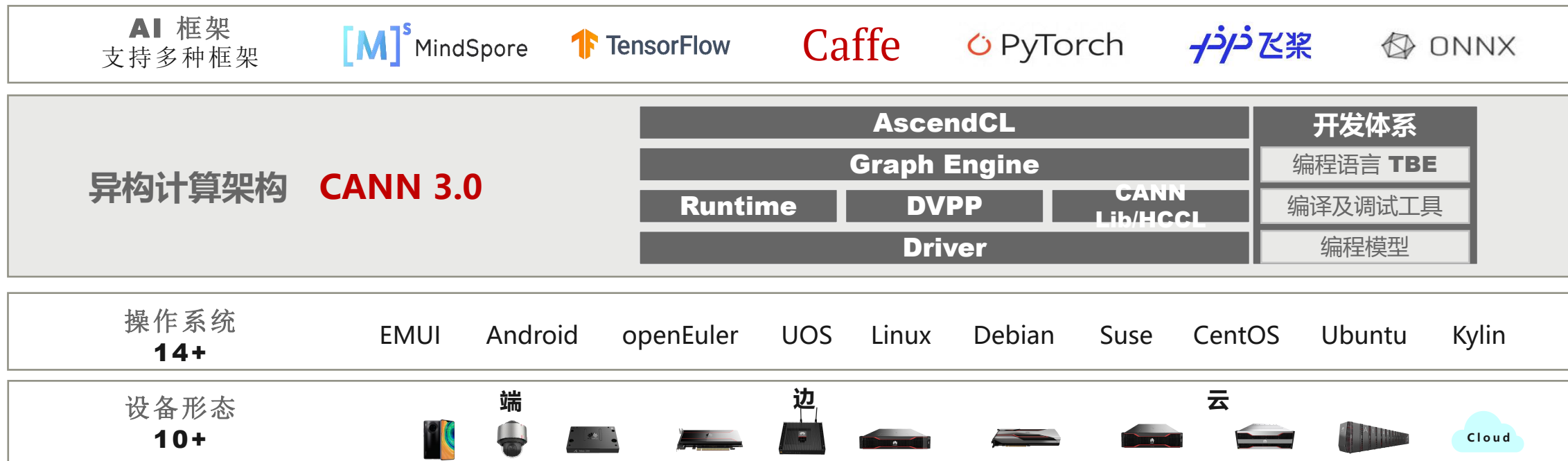
5 200DK外设接口相关



昇腾：构筑业界最强AI算力平台，使能千行百业智能化转型



CANN3.0：端边云协同，使能全场景AI开发



端边云全场景协同

10+ 设备形态
14+ 操作系统
多种AI框架



AscendCL使能高效开发

统一API，后向兼容
四大开放性设计
两种算子开发方式



释放硬件澎湃算力

极致的图编译技术
丰富的算子库支持，1000+高性能算子

能力分层开放，使能行业AI应用



**驱动、编译器、软件栈、工具链
及基础服务**



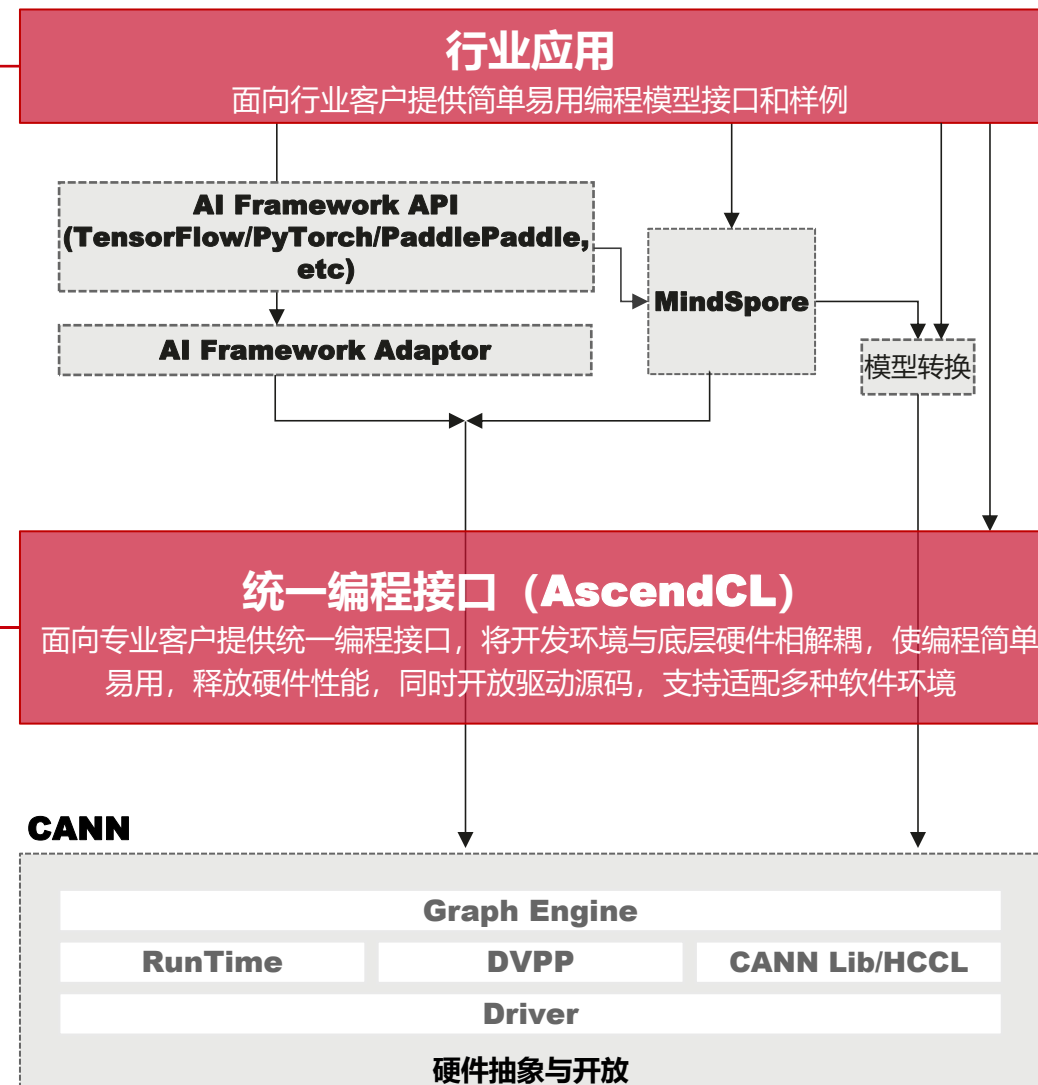
昇腾芯片



智能边缘



数据中心



目录

TABLE OF CONTENTS

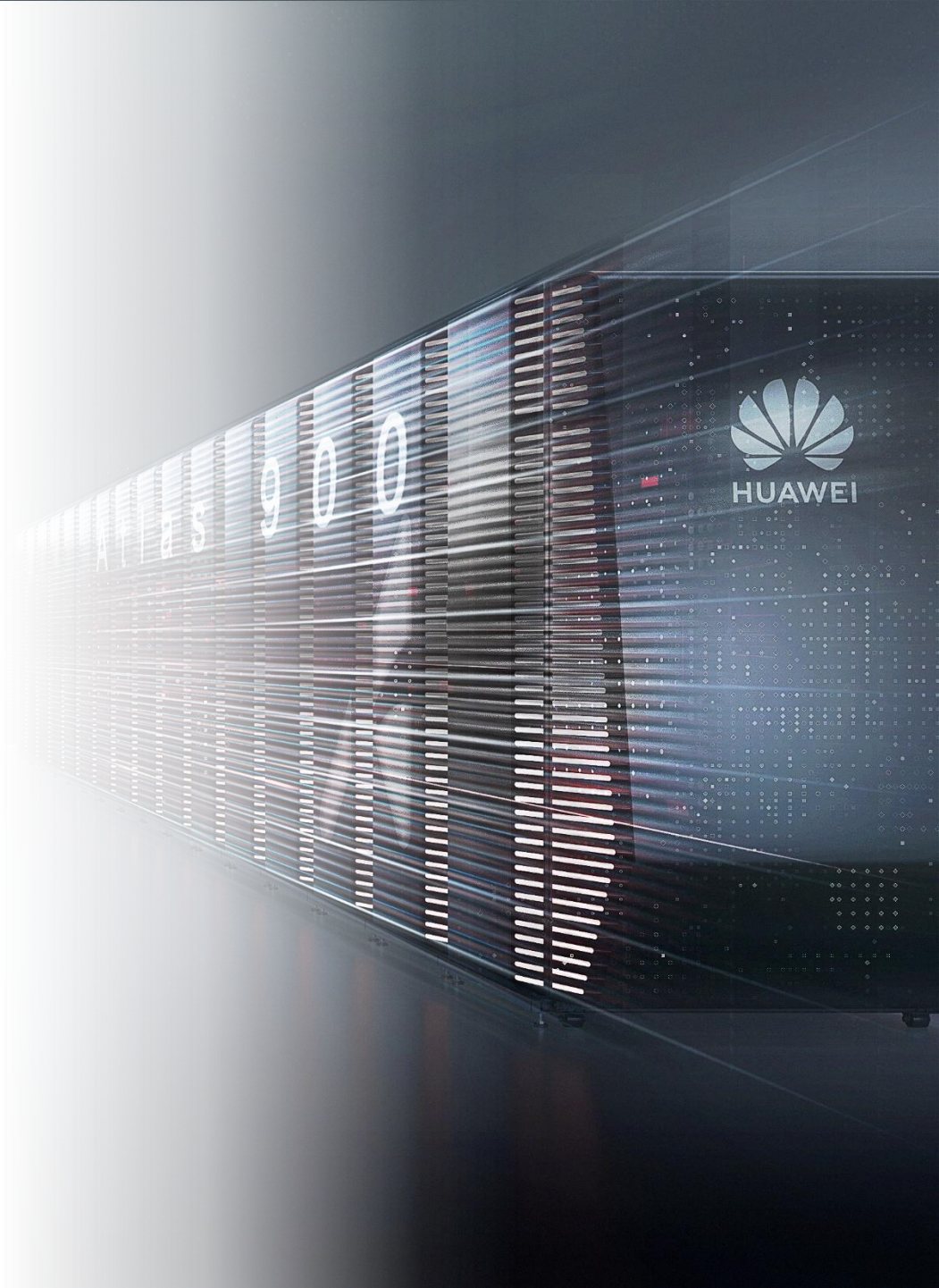
1 异构计算架构 CANN

2 CANN AI 应用开发流程

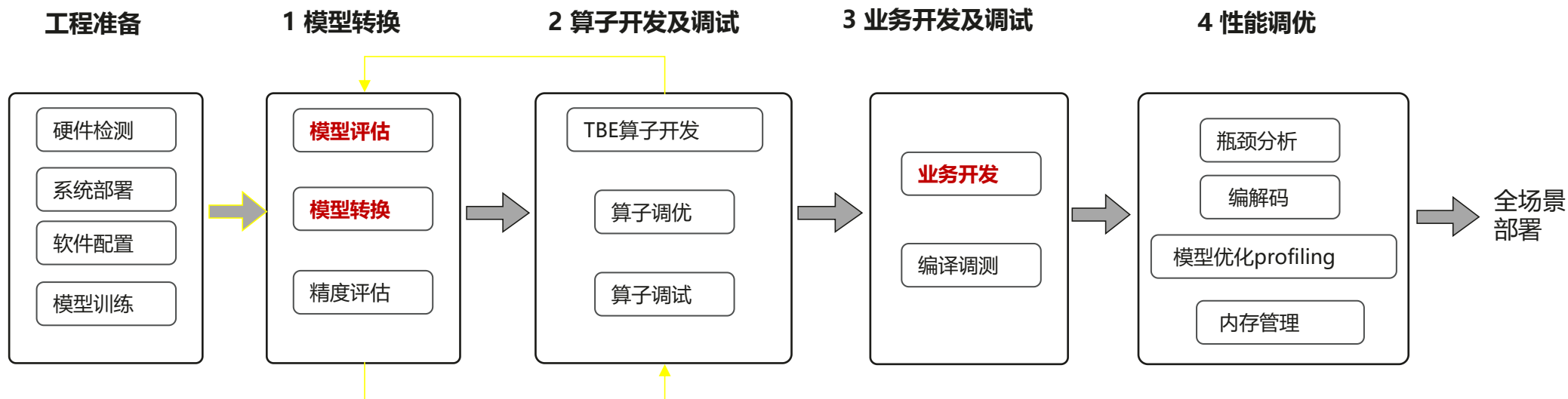
3 实战案例解析

4 性能&精度

5 200DK外设接口相关



CANN AI 应用开发流程



0) 工程准备

硬件：服务器及推理卡主备就绪，安装操作系统，配置网络。

训练后的模型：MindSpore、TensorFlow、pytorch、caffe等模型；

1) 模型转换

离线模型：ATC转换工具

2) 算子开发及调试

自定义算子开发：TBE DSL，TIK等算子开发工具。

3) 业务开发及调试

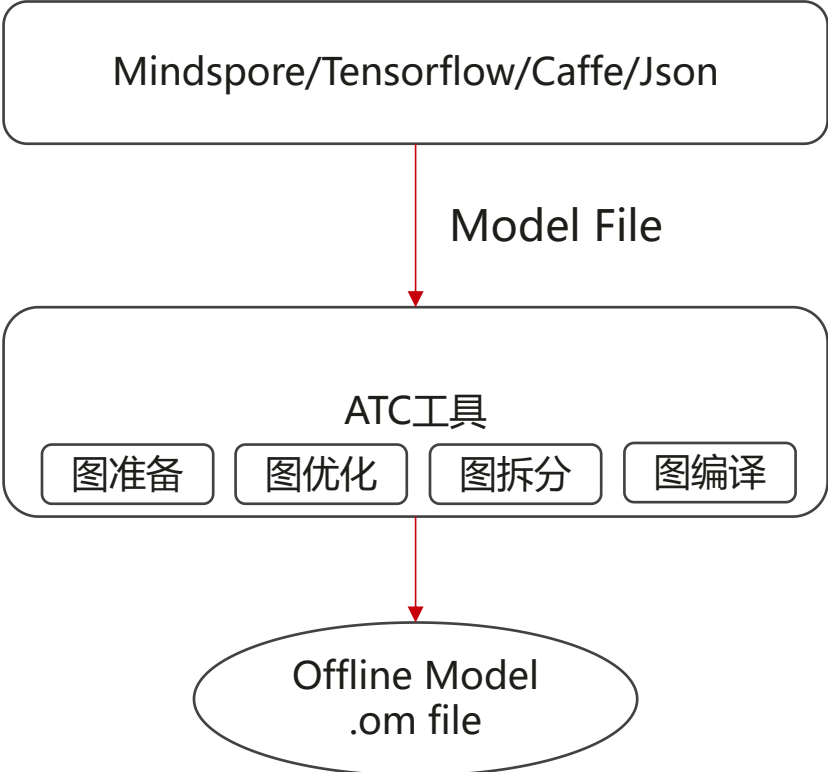
ACL接口：资源初始化，数据传输，**数据预处理（DVPP/AIPP等）**，**模型推理**，**数据后处理**等。

4) 性能调优

性能优化：瓶颈分析，内存优化，模型优化等

ATC模型转换

ATC工具将开源框架的网络模型（如Mindspore、Caffe、TensorFlow等）以及单算子Json文件转换成昇腾AI处理器支持的离线模型，模型转换过程中可以实现算子调度的优化、权值数据重排、内存使用优化等，可以脱离设备完成模型的预处理。



参数名称	参数描述
--mode	运行模式。 <ul style="list-style-type: none">0: 生成适配昇腾AI处理器的离线模型。1: 离线模型或模型文件转json。3: 仅做预检，检查模型文件的内容是否合法。5: ptxt格式文件转json。
--model	原始模型文件路径。
--weight	权重文件路径。当原始模型是Caffe时需要指定。
--framework	原始框架类型。0: Caffe; 1: MindSpore; 3: TensorFlow
--output	<ul style="list-style-type: none">如果是开源框架的网络模型：存放转换后的离线模型的路径以及文件名，例如： \$HOME/test/out/caffe_resnet18或\$HOME/test/out/tf_resnet18，转换后的模型以“.om”后缀结尾。如果是单算子json文件：存放转换后的单算子模型的路径，例如：\$HOME/test/out/op_model。
--input_format	输入数据格式。支持NCHW、NHWC、ND三种格式。
--out_nodes	指定输出节点。
--input_shape	模型输入数据的shape。
--json	当mode为1时必填。离线模型或模型文件转换为json格式文件的路径和文件名，例如： \$HOME/test/out/resnet18.json。
--om	当mode为1时必填。需要转换为json格式的离线模型或模型文件的路径，例如： <ul style="list-style-type: none">Caffe: \$HOME/test/resnet18.prototxt或\$HOME/test/out/caffe_resnet18.om。TensorFlow: \$HOME/test/resnet18_tensorflow.pb或\$HOME/test/out/tf_resnet18.om。
--insert_op_conf	输入预处理算子的配置文件路径，例如aipp算子。
--soc_version	模型转换时指定芯片版本，例如：Ascend310。

模型转换示例

训练模型

参数导出

参数固化

模型转换

```
# =====Training a model from scratch.=====
DATASET_DIR=/tmp/data/flowers/
CHECKPOINT_DIR=/tmp/train_logs
python train_image_classifier.py \
  -----train_dir=${CHECKPOINT_DIR} \
  -----dataset_name=flowers \
  -----dataset_split_name=train \
  -----dataset_dir=${DATASET_DIR} \
  -----model_name=inception_v3 \
  -----clone_on_cpu=True \
  -----save_summaries_secs 5 \
  -----save_interval_secs 20
```

```
# ----- (Way1) Script -----
echo "=====
echo "Freezing the exported Graph"
echo "=====

python freeze_graph.py \
  -----input_checkpoint=${CHECKPOINT_DIR}/model.ckpt-53 \
  -----output_graph=model_test/frozen_inception_v3.pb \
  -----output_node_names="InceptionV3/Predictions/Reshape_1"
```

Mindspore	atc --input_format=NCHW --framework=1 --model= resnet50-2_1.geir --input_shape="x:1,3,224,224" --output=bs_1_googlenet --soc_version=Ascend310 --disable_reuse_memory=1
-----------	---

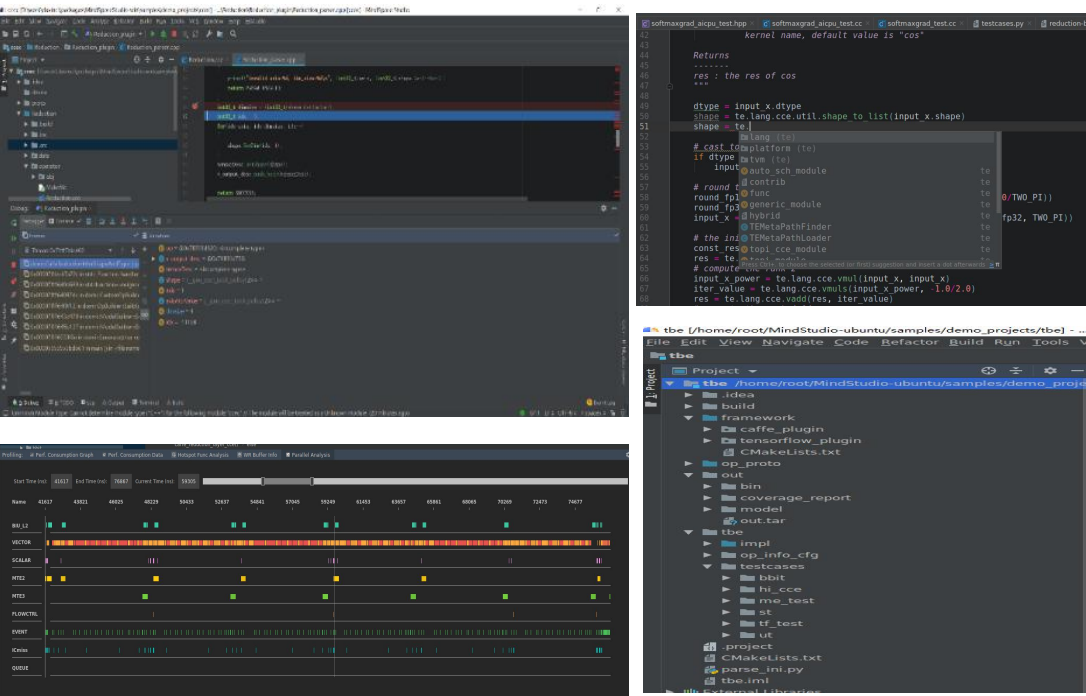
TensorFlow	atc --model=\$HOME/test/resnet18_tensorflow.pb --framework=3 --output=\$HOME/test/out/tf_resnet18 --soc_version=Ascend310
------------	---

Caffe	atc --model=\$HOME/test/resnet50.prototxt --weight=\$HOME/test/resnet50.caffemodel --framework=0 --output=\$HOME/test/out/caffe_resnet50 --soc_version=Ascend310
-------	--

自定义算子开发工具TBE

TBE(Tensor Boost Engine)工具是一款面向Davinci NPU的算子开发工具，在TVM(Tensor Virtual Machine)框架基础上扩展，提供了一套Python API接口来开发NPU算子。

TBE开发IDE工具



- 支持算子开发(语法高亮、代码自动补全等)
- 支持性能Profiling
- 支持单元测试框架

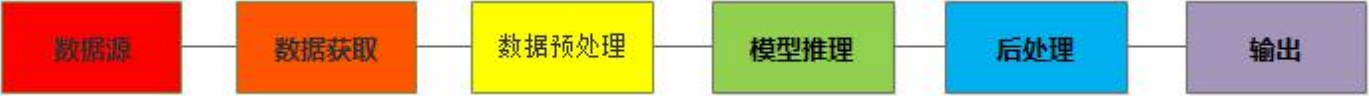
多种开发方式选择

开发方式	使用场景
TBE DSL	开发难度： 易 适用人群： 入门开发者， 仅需要了解NN和TBE DSL相关知识 特点： TBE工具提供自动优化机制， 给出较优的调度流程
TVM原语	开发难度： 中 适用人群： 中级开发者， 还需要了解NN、TVM 架构及原语、Davinci硬件相关知识 特点： 开发者可自己控制算子的调度流程， 但不需要深入掌握Davinci指令的使用
TIK	开发难度： 难 适用人群： 高级开发者， 还需要掌握Davinci硬件Buffer架构， 对性能有较高要求 特点： 接口偏底层， 用户需要自己控制数据流以及算子的调度流程

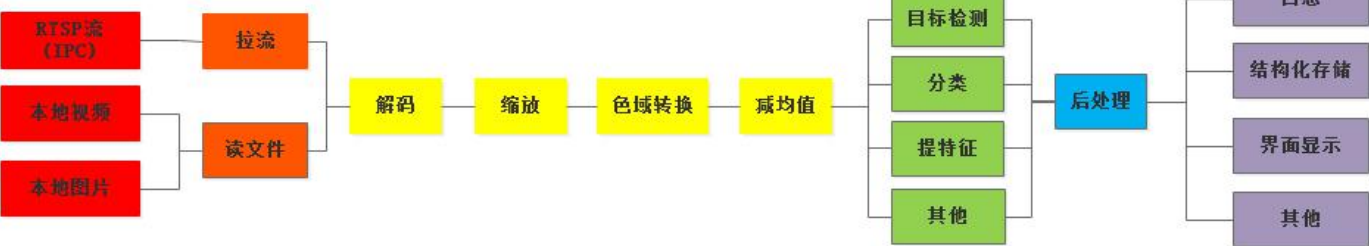
- 多种开发方式结合，兼顾开发效率和开发灵活性
- 内外部都采用TBE工具，利于生态构建

推理应用开发流程

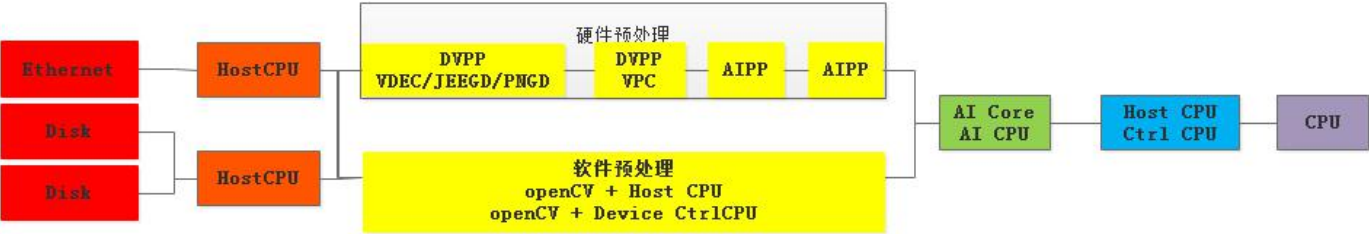
推理业务流程



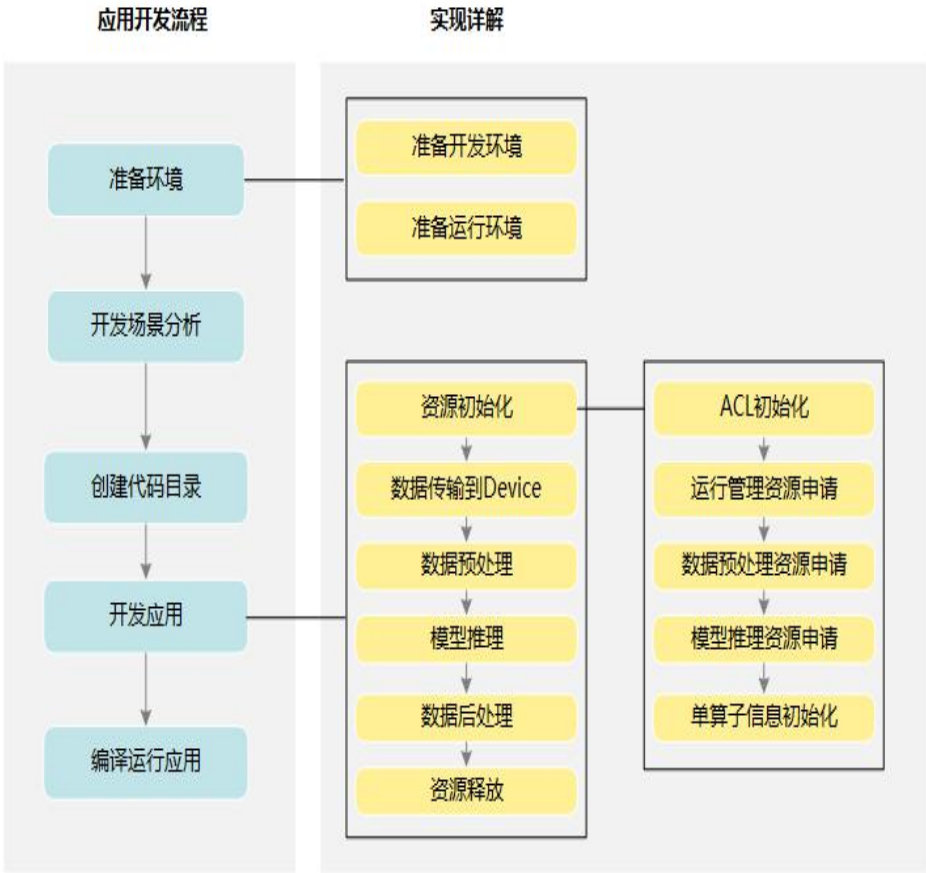
软件模块图



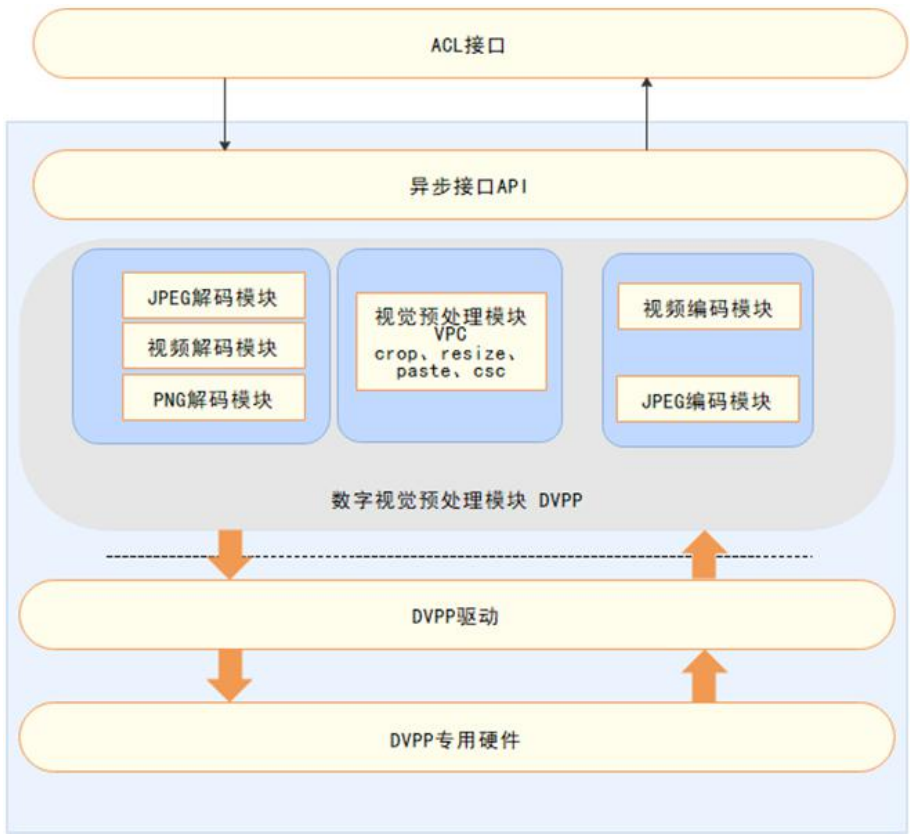
硬件模块图



开发流程介绍



数据预处理之DVPP



受网络结构和训练方式等因素的影响，绝大多数神经网络模型对输入数据都有格式上的限制。在计算视觉领域，这个限制大多体现在图像的尺寸、色域、归一化参数。

昇腾软件栈提供了两套专门用于数据预处理的工具，其中一套叫做AIPP；另一套叫做DVPP（Digital Vision Preprocessing），也就是数字视觉预处理模块。

接口	英文全称	中文名称	功能
VPC	Vision Preprocess Core	视觉预处理模块	支持对图片做抠图、缩放、叠加、拼接、格式转换等操作
JPEGD	JPEG Decode	JPEG图片解码	实现.jpg、.jpeg、.JPG、.JPEG图片的解码，对于硬件不支持的格式，会使用软件解码
JPEGE	JPEG Encoder	JPEG图片编码	将YUV格式图片编码成.jpg图片，支持YUV422 Packed、YUV420SP (NV12,NV21)
PNGD	Portable Network Graphics Decoder	便携式网络图像格式解码	实现PNG格式图片的硬件解码。支持RGBA、RGB格式的图片解码成RGBA、RGB格式
VDEC	Video Decoder	视频解码	支持H264、H265两种视频格式的解码
VENC	Video Encoder	视频编码	实现YUV/YVU420图片数据的编码，支持H264、H265两种视频格式的编码

数据预处理之AIPP

AIPP (AI Preprocessing)：用于在AI Core上完成图像预处理，包括色域转换（转换图像格式）、图像归一化（减均值/乘系数）和抠图（指定抠图起始点，抠出神经网络需要大小的图片）等。

AIPP区分为静态AIPP和动态AIPP。您只能选择静态AIPP或动态AIPP方式来处理图片，不能同时配置静态AIPP和动态AIPP两种方式。

静态AIPP：模型转换时设置AIPP模式为静态，同时设置AIPP参数，模型生成后，AIPP参数值被保存在离线模型 (*.om) 中，每次模型推理过程采用固定的AIPP预处理参数（无法修改）。如果使用静态AIPP方式，多Batch情况下共用同一份AIPP参数。

动态AIPP：模型转换时仅设置AIPP模式为动态，每次模型推理前，根据需求，在执行模型前设置动态AIPP参数值，然后在模型执行时可使用不同的AIPP参数。设置动态AIPP参数值的接口请参见设置动态AIPP参数。如果使用动态AIPP方式，多Batch可使用不同的AIPP参数。

目录

TABLE OF CONTENTS

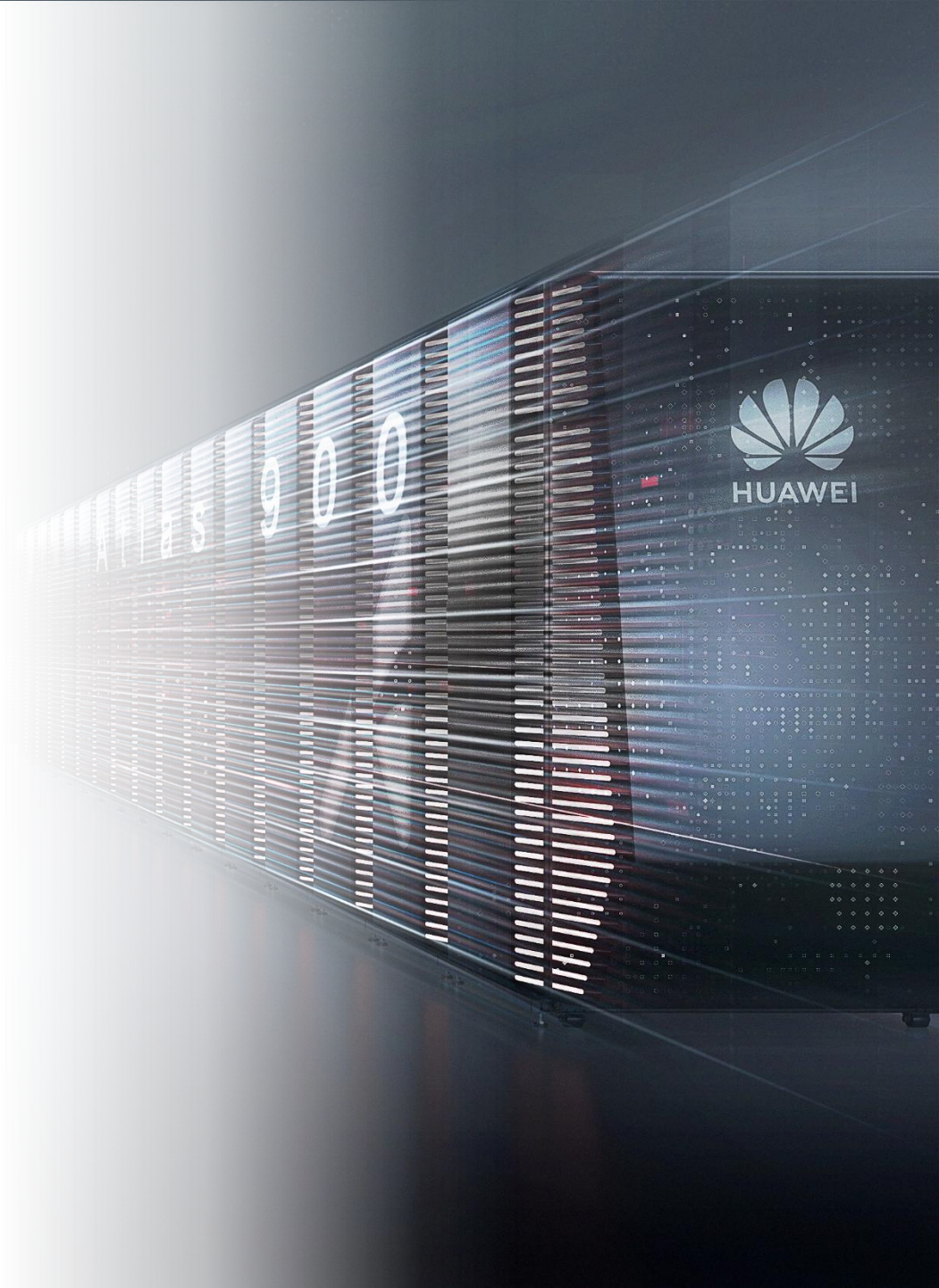
1 异构计算架构 CANN

2 CANN AI 应用开发流程

3 实战案例解析

4 性能&精度

5 200DK外设接口相关



ACL高层封装 – 开源代码

C++版本: <https://gitee.com/ascend/samples/tree/master/cplusplus/common/atlasutil>

python版本: https://gitee.com/ascend/samples/tree/master/python/common/atlas_utils

gitee

开源软件 企业版 特选 高校版 博客 我的

搜开源

AtlasVideoCapture

方法: AtlasVideoCapture(uint32_t cameraId, uint32_t width = 1280, uint32_t height = 720, uint32_t fps = 20)

说明: 在Atlas200DK上打开指定槽位的摄像头。如果该摄像头不可用, 只生成实例, 不会打开摄像头

输入参数: cameraId: 摄像头id, 0 表示CAMERA0槽位的摄像头, 1 表示CAMERA1槽位的摄像头

width:摄像头分辨率宽

height:摄像头分辨率高

fps:帧率, 参数范围为[1, 20]

返回值: 无

约束: 1. 只支持atlas200dk;

2. 摄像头默认分辨率参数设置需要符合驱动要求, 当前支持5种分辨率: 1920 x 1080, 1280 x 720, 704 x 576, 704 x 288, 352 x 288。

AtlasVideoCapture

方法: AtlasVideoCapture(const string& videoPath, aclrtContext context = nullptr)

说明: 解码视频videoPath

输入参数: videoPath: 视频文件或者rtsp地址;

context: 解码器使用dvpp vdec解码时使用的acl context。默认情况下使用当前线程的context

返回值: 无

约束: 解码器使用ffmpeg+vdec解码视频, 在创建实例前需要初始化acl(aclInit)和设置device(aclrtSetDevice)

注意事项: 无

IsOpened

atlasutil库使用说明

编译方法

第三方库依赖

编译步骤

部署方法

接口说明

AtlasCapture类

AtlasVideoCa...

AtlasVideoCa...

IsOpened

Get

Set

Read

Close

DvppProcess类

DvppProcess

InitResource

Resize

JpegD

JpegE

AtlasModel类

AtlasModel

Init方法

CreateInput

CreateInput

CreateInput

DestroyInput

Execute方法

master

samples / python / common / atlas_utils

新建文件 新

wuyan fixed crowd_count 4173bb2 27天前

lib fix bug

presenteragent modify

README.md update python/common/atlas_utils/README.md.

__init__.py python样例开发公共库

acl_dvpp.py fixed crowd_count

acl_image.py python video decode implement with pyav

acl_logger.py python video decode implement with pyav

acl_model.py python video decode implement with pyav

acl_resource.py update python/common/atlas_utils/acl_resource.py.

camera.py python video decode implement with pyav

chanel_id_generator.py python video decode implement with pyav

constants.py python video decode implement with pyav


dvpp_vdec.py python video decode implement with pyav

resource_list.py python video decode implement with pyav

utils.py python video decode implement with pyav

video.py python video decode implement with pyav

15 Huawei Confidential



实战案例解析 – 黑白图像上色

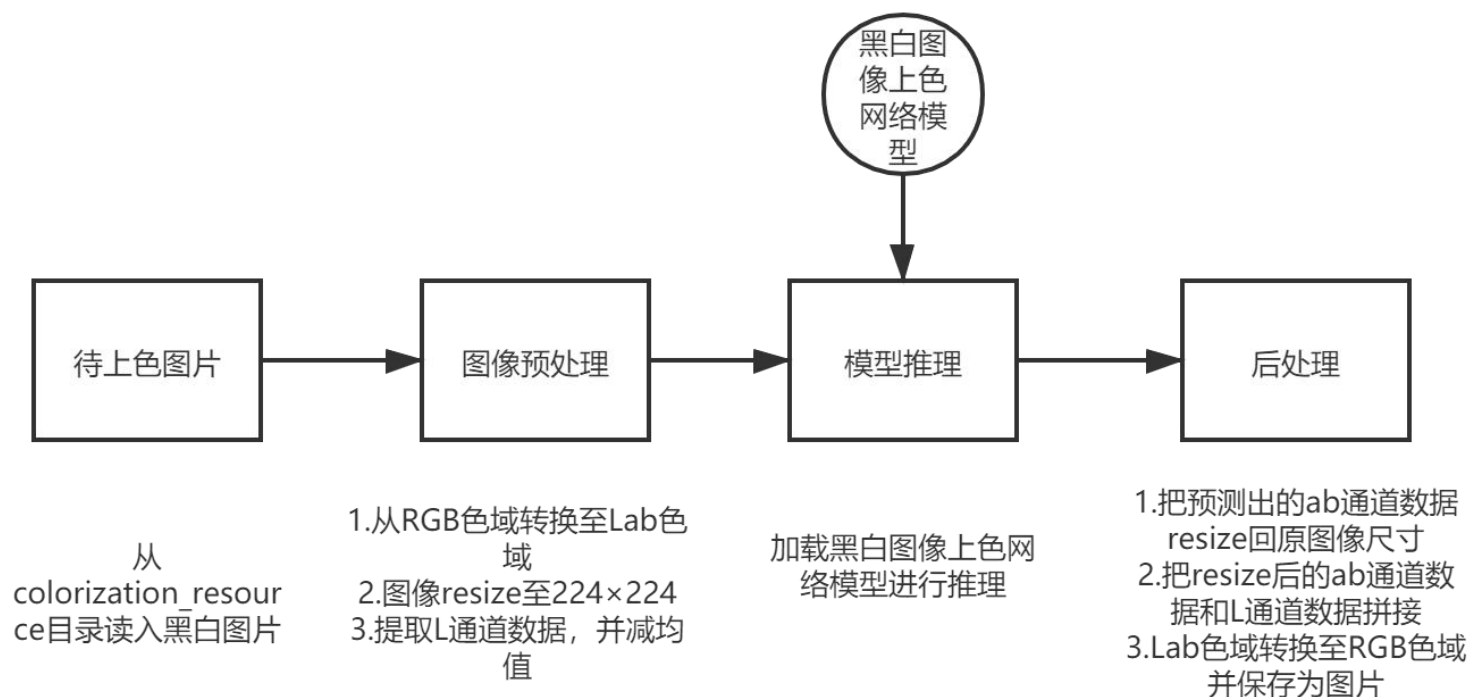
在线体验: <https://www.hiascend.com/zh/developer/mindx-sdk/cartoon/990674866img?fromPage=1>

开源案例: https://gitee.com/ascend/samples/tree/master/cplusplus/level2_simple_inference/6_other/colorization

通过源测试脚本研究模型推理过程: <https://github.com/richzhang/colorization/blob/master/colorization/colorize.py>

预处理过程: RGB格式读入转Lab, resize到224*224, 提取L通道, 减均值 (-50)

后处理过程: 推理的结果ab通道, resize到224*224, 与输入L合并为Lab, 转RGB, 保存为jpeg图片



实战案例解析 – 模型转换&快速评估

Choose Model ☒ Configure Input ☒ Output ☒ Configure Data Pre-Processing ☒

Image Pre-processing ☒

Configuration (data_l) ☒

Input Format: YUV420 sp BT.601(Video Rang...)

Source Image Size: H 224 W 224

Model Image Format: ☐

Need Crop: ☐

Need Data Normalization: ☒

Conversion Type: ☒ UINT8->FLOAT16

Mean:	R	50	G	50	B	50
Min:	R	0.0	G	0.0	B	0.0
Variance:	R	1.0	G	1.0	B	1.0

Choose Model ☒ Configure Input ☒ Output ☒ Configure Data Pre-Processing ☒

Image Pre-processing ☐

Configuration (data_l) ☐

Input Format: YUV420 sp BT.601(Video Rang...)

Source Image Size: H 224 W 224

Model Image Format: ☐

Need Crop: ☐

Need Data Normalization: ☐

Conversion Type: ☒ UINT8->FLOAT16

快速评估工具: <https://gitee.com/ascend/tools/tree/master/msame>

```
./msame --model "/home/HwHiAiUser/msame/colorization.om" --output
```

```
"/home/HwHiAiUser/msame/out/" --outfmt TXT --loop 1
```


目录

TABLE OF CONTENTS

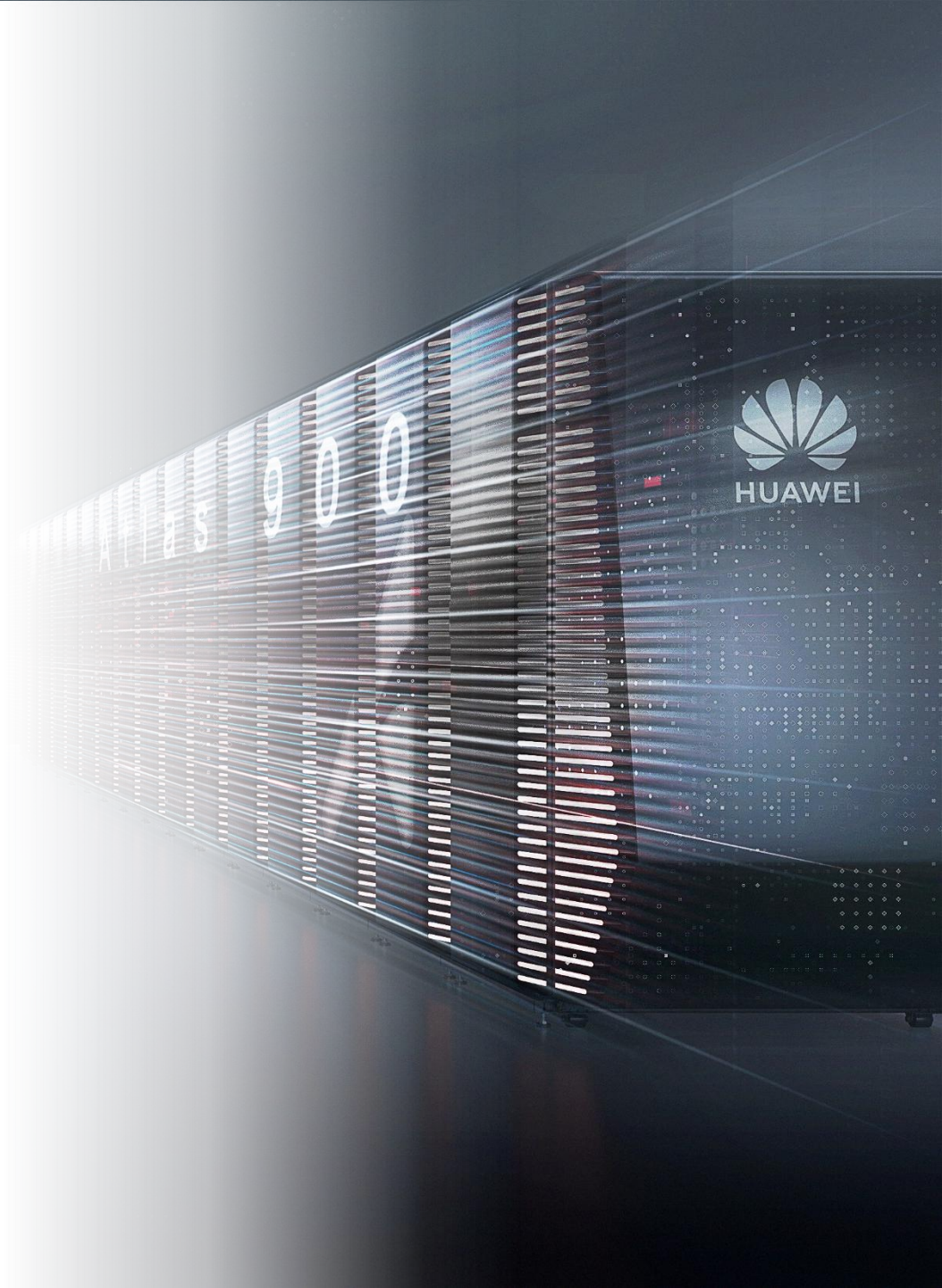
1 异构计算架构 CANN

2 CANN AI 应用开发流程

3 实战案例解析

4 性能&精度

5 200DK外设接口相关



性能&精度

1. 使用DVPP和AIPP做预处理;
2. 使用单算子调用处理前处理或者后处理性能瓶颈;
3. 多线程并行处理前处理/推理/后处理;
4. 昇腾模型压缩工具;
5. 多机多卡并行计算;

各种工具集合:

https://support.huaweicloud.com/auxiliarydevtool-cann502alpha5infer/atlasinfertool_16_0002.html

目标检测应用开发 - 模型获取

原始模型网络链接地址:

<https://github.com/weiliu89/caffe/tree/ssd>

原始模型测试脚本:

https://github.com/weiliu89/caffe/blob/ssd/examples/ssd/ssd_detect.py

```
def detect(self, image_file, conf_thresh=0.5, topn=5):
    ...
    SSD detection
    ...
    # set net to batch size of 1
    # image_resize = 300
    self.net.blobs['data'].reshape(1, 3, self.image_resize, self.image_resize)
    image = caffe.io.load_image(image_file)

    #Run the net and examine the top k results
    transformed_image = self.transformer.preprocess('data', image)
    self.net.blobs['data'].data[...] = transformed_image

    # Forward pass.
    detections = self.net.forward()[0]

    # Parse the outputs.
    det_label = detections[0,0,:,1]
    det_conf = detections[0,0,:,2]
    det_xmin = detections[0,0,:,3]
    det_ymin = detections[0,0,:,4]
    det_xmax = detections[0,0,:,5]
    det_ymax = detections[0,0,:,6]
```



预处理过程:

加载图像数据->转float32->NCHW->BGR->**resize(300,300)**->减均值[104, 117, 123]



后处理:

[1, 1, 200, 7]

目标检测应用开发 - 预处理方案

基于对原始模型的理解，以及对DVPP和AIPP的了解，在图像预处理时DVPP和AIPP如下分工：

DVPP:

1. 解码：JPEG图片先解码为YUV420SP（输出128*16对齐） 800*600 -> 896*608
2. 图像缩放：cropandpast (300,304) (16*2对齐) (896*608(800*600)->300*304(300*300))
3. 输出数据类型：uint8

AIPP:

1. 色域转换：YUV->BGR
2. 减均值
3. 抠出Crop (0, 0, 300,300)
4. 图像数据类型转换：Uint8->FP16（达芬奇推理的要求，模型最后实际的输入是fp16）

目标检测应用开发 – 模型转换

Model Converter

Choose Model Configure Input Output Configure Data Pre-Processing

* Model File: AscendProjects/sample-objectdetection/caffe_model/vgg_ssd.prototxt

* Weight File: cendProjects/sample-objectdetection/caffe_model/vgg_ssd.caffemodel

Model Name: vgg_ssd

Target SOC Version: Ascend310

Model Converter

Choose Model Configure Input Output Configure Data Pre-Processing

Input Type: FP32

Input Format: ☒ NCHW ☐ NHWC

Input Node: data

N: 1 C: 3 H: 300 W: 300

Output Type: FP32

Output For...: Select Output Nodes

Model Converter

Choose Model Configure Input and Output Configure Data Pre-Processing

Data Preprocessing: ☐

Input Node: (data) ☒ 图像输入格式选择为YUV420 sp

Input Image Format: YUV420 sp BT.601 (Video Ran...)

Input Image Resolution: H 300 W 304 此处填如304, 意为输入om模型的图片宽度是基于原始模型要求输入宽度向上取16的最小倍数

Model Image Format: ☒ BGR

Crop: ☐ 当输入的图像数据格式与模型要求的格式不同时需要开启此选项

抠图开始位置为图像左上角

抠图区域不支持修改

Cropping Start: H 0 W 0

Cropping Area: H 300 W 300

Data Normalization: ☐

Conversion Type: ☒ UINT8->FP16

Mean	R	104	G	117	B	123	减均值为默认
Min	R	0.0	G	0.0	B	0.0	
Variance	R	1.0	G	1.0	B	1.0	

Load Configuration Previous Next Cancel Finish

高清图像修复 – 大矩阵乘解决自注意力矩阵计算问题

<https://www.hiascend.com/zh/developer/mindx-sdk/imageInpainting?fromPage=1>

AI 图像修复

 gitee

在线实验 →

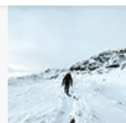
AI 修复体验

概述

图像预处理

模型推理结果后处理

效果展示



上传图片

注：仅支持PNG、JPEG、BMP图片格式



 框选工具 (2)

立即修复

目录

TABLE OF CONTENTS

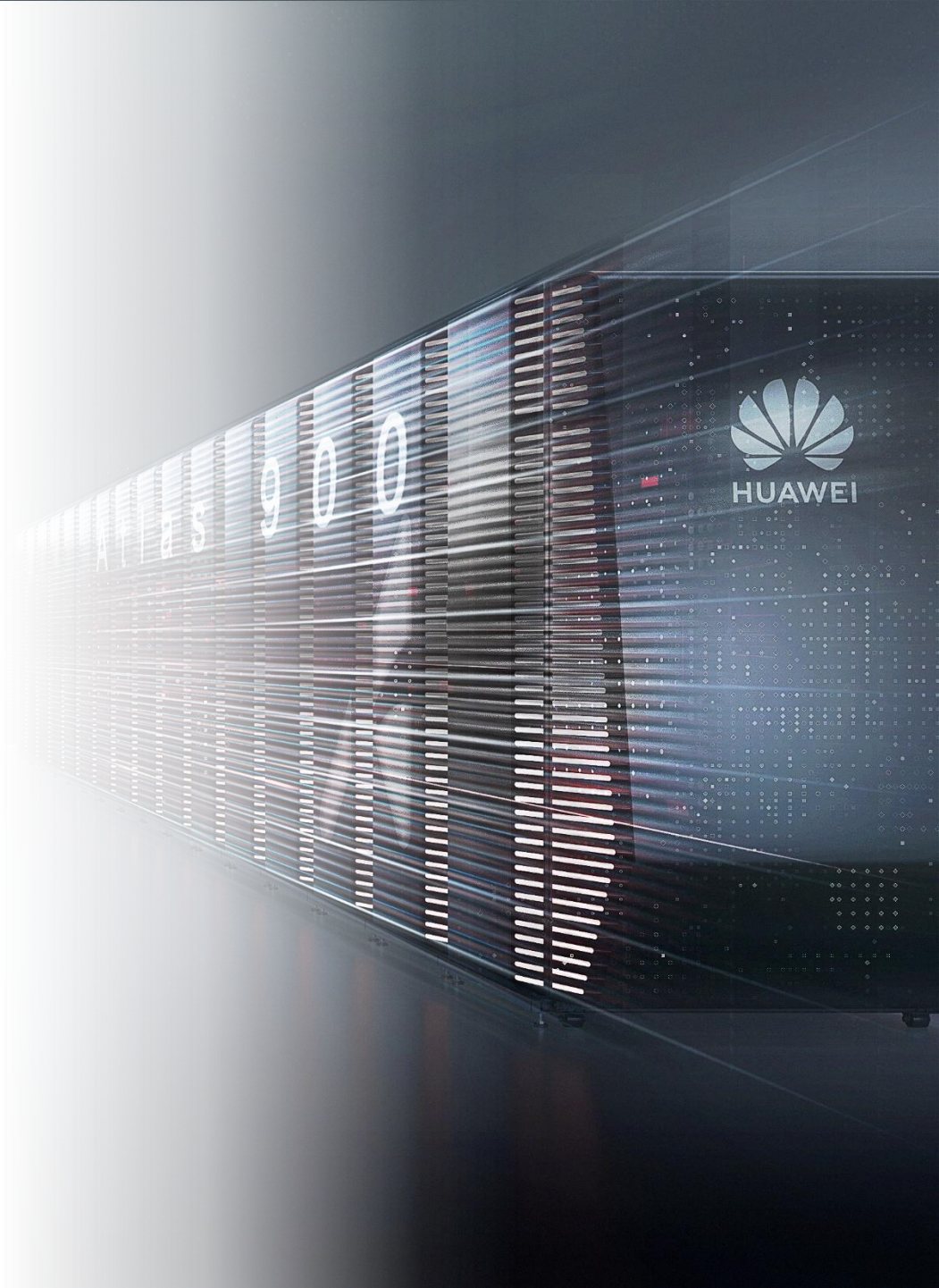
1 异构计算架构 CANN

2 CANN AI 应用开发流程

3 实战案例解析

4 性能&精度

5 200DK外设接口相关



200DK外设接口

https://gitee.com/ascend/samples/wikis/Atlas200DK%E5%A4%96%E8%AE%BE%E4%B8%80%E6%8C%87%E7%A6%85?sort_id=3541317

- 1、开发环境软硬件配置
- 2、Atlas200DK_GPIO
- 3、Atlas200DK串口
- 4、I2C

应用案例实战 – 更多

访问Ascend社区:

<https://www.hiascend.com/>

了解更多案例: 开发者 --》社区项目 <https://www.hiascend.com/developer/case-studie>

应用领域

全部

图像分类

目标检测

图像处理

检测+分类

姿态识别

人脸识别

图像生成

三维重建

图像增强

机器人

NLP+CV

AR 阴影生成

实例分割

NLP

模式识别

医学图形

通用

特征检索

OCR

适用产品

全部

Atlas 200 DK

Atlas 300I

Atlas 500

Atlas 500 Pro


Atlas 800

接口类型

全部


MindX SDK

AscendCL




在线体验

密集人群统计




在线体验

昇腾智能问答系统




在线体验

昇腾 CANN 智春联




在线体验

AI 图像修复



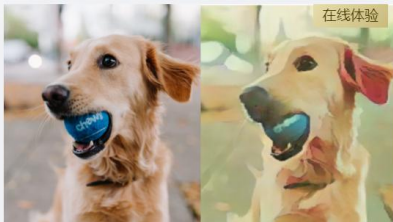
在线体验

垃圾分类




在线体验

AI 风景画



在线体验

卡通图像生成



在线体验

乐府作诗

Thank you.



昇腾开发者社区

<http://hiascend.com>

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

**Bring digital to every person, home, and
organization for a fully connected,
intelligent world.**

**Copyright©2020 Huawei Technologies Co., Ltd.
All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

