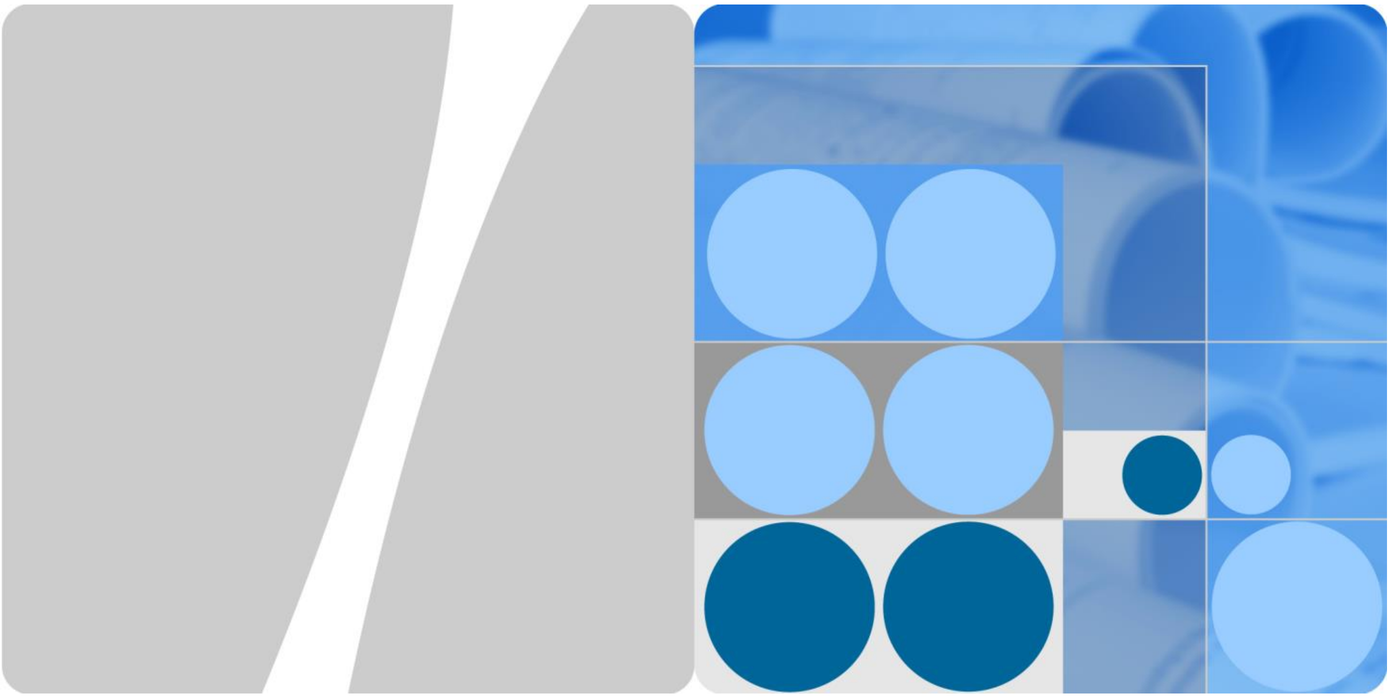


HiAI DDK V320

Quick Start

Issue 04

Date 2020-02-28

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei HiAI Application

Send an application email to developer@huawei.com.

Email subject: HUAWEI HiAI + Company name + Product name

Email body: Cooperation company + Contact person + Phone number + Email address

We will reply to you within 5 working days.

Official website: https://developer.huawei.com/consumer/en/

About This Document

Purpose

This document provides an overview of Huawei HiAI DDK V320 and the integration process.

This document is used in conjunction with the following documents.

Document Name	Description
Huawei HiAI DDK V320 Release Notes	Describes the version changes and feature updates in HiAI DDK V320.
Huawei HiAI DDK V320 FAQs	Describes FAQs related to the HiAI DDK.
Huawei HiAI DDK V320 IR Model Building Instructions	Describes the model building methods and APIs. Operators under frameworks other than TensorFlow and Caffe are also supported.
Huawei HiAI DDK V320 Quick Start	Describes the HiAI DDK.
Huawei HiAI DDK V320 Model Inference and Integration Instructions	Describes the model integration and compilation methods and integration APIs.
Huawei HiAI DDK V320 OMG Tool Instructions	Describes the usage of the OMG tool.
Huawei HiAI DDK V320 Lightweight Tool Instructions	Describes how to use the lightweight tool.
Huawei HiAI DDK V320 Operator Specifications	Describes the restrictions of operators supported by HiAI DDK V320.
Huawei HiAI DDK V320 Acronyms and Abbreviations	Describes the acronyms, abbreviations, and terms in the HiAI DDK.
Huawei HiAI DDK V320	Describes how to use the system debug tool.

Document Name	Description
System Debug Tool Instructions	

Change History

Date	Version	Change Description
2020-02-28	04	Added the description of the system debug tool and general-purpose Arm processor.
2019-12-31	03	Added the description of HiAI DDK V320.
2019-11-18	02	Added a prompt when the AIPP function is enabled in the demo.
2019-09-06	01	Added the description of HiAI DDK V310.

Contents

Figures

Tables

1 Introduction

HiAI is an artificial intelligence (AI) computing platform for mobile devices. The HiAI device development kit (DDK) is a HiAI resource package open to third-party developers.

HiAI APIs constitute an AI computing library of a mobile computing platform, enabling developers to efficiently compile AI apps that can run on mobile devices.

The HiAI APIs are released as unified binary files. They are used to accelerate neural network computing through the HiAI heterogeneous computing platform. Currently, the HiAI APIs can run only on the Kirin SoC.

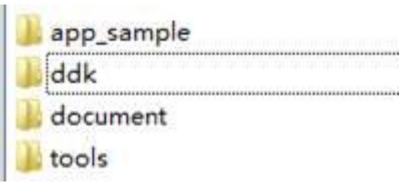
The HiAI APIs are integrated into the Android system that uses the Kirin SoC. Developers can run the neural network model in the integrated environment and invoke HiAI APIs to accelerate computing.

2 DDK Description

2.1 Overview

The device development kit (DDK) is an open HiAI development package. Figure 2-1 shows a complete HiAI DDK.

Figure 2-1 DDK directory structure



app_sample
ddk
document
tools

- **app_sample** stores the source code of Android demo apps.
- **ddk** stores the open HiAI SDK
- **document** stores development reference files.
- **tools** stores the offline model generator (OMG) tool for converting Caffe/TensorFlow models and the lightweight tool.

NOTE

IR_model_demo in the **app-sample** directory can only run on smartphones powered by Huawei-developed NPU.

2.2 Directory sample

In **sample\inference_npu_demo\Demo_Source_Code** provides a series of sample codes for input pre-processing, model loading, model forward computation, forward computation result post-processing, model unloading, and time statistics collection by using the SqueezeNet classification network model (https://github.com/DeepScale/SqueezeNet/tree/master/SqueezeNet_v1.1) as an example. It also offers sample codes for synchronous and asynchronous modes. The AIPP feature provided by V320 supports input pre-processing and is supported only by Kirin 990. The code for loading the AIPP model in the sample code is commented out by default. You can enable the feature as required by setting the **initModels()** function of class **MainActivity** in the **Demo_Soure_Code\app\src\main\java\com\huawei\hiaidemo\view\MainActivity.java** file in the SDK.

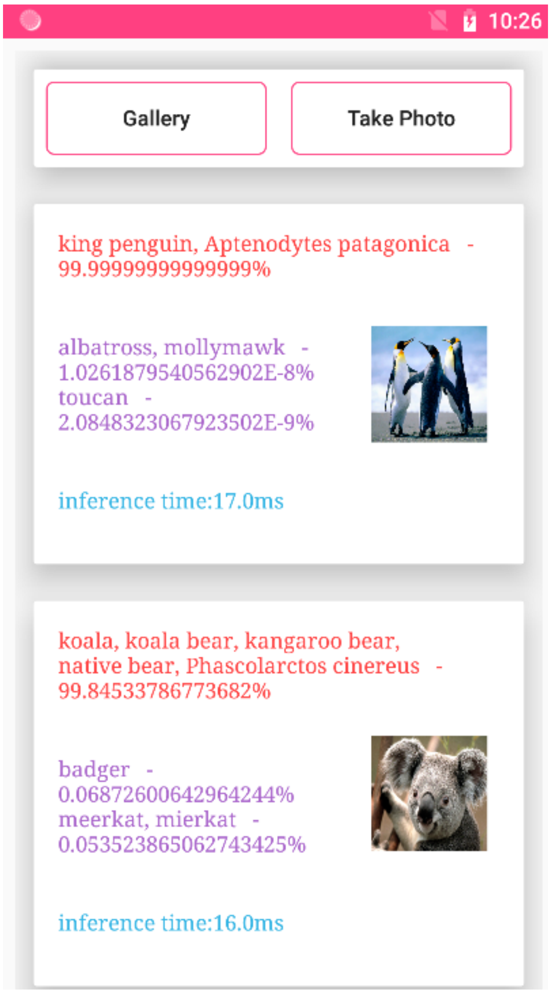
```
ModelInfo model_1 = new ModelInfo();
model_1.setModelSaveDir(path);
model_1.setUseAIPP(true);
model_1.setOfflineModel("hiai.om");
model_1.setOfflineModelName("hiai");
model_1.setOnlineModelLabel("labels_caffe.txt");
demoModelList.add(model_1);
```

NOTE

Use Android Studio 2.2 or a later version, which can be downloaded from https://developer.android.com/studio/index.html.

Import and execute the source code in **sample**. The app supports **Gallery** and **Take Photo**, which allow you to select images from the gallery and take photos, respectively. Figure 2-2 shows the app demo UI.

Figure 2-2 Running effect of the app demo UI



king penguin, Aptenodytes patagonica -
99.99999999999999%

albatross, mollymawk -
1.0261879540562902E-8%
toucan -
2.0848323067923502E-9%

inference time:17.0ms

koala, koala bear, kangaroo bear,
native bear, Phascolarctos cinereus -
99.84533786773682%

badger -
0.06872600642964244%
meerkat, mierkat -
0.053523865062743425%

inference time:16.0ms

2.3 Directory ddk

The **ai_ddk_lib** folder in **ddk** contains the following two types of files:

- Model inference: dependency library and related header file

Directory	Description
ai_ddk_lib\lib64\libhiai.so	Dynamic library required for the DDK to use the NPU for model inference
ai_ddk_lib\lib64\libhcl.so	Dynamic library required for the DDK to use the NPU for model inference
ai_ddk_lib\lib64\libcpucl.so	(optional) Dynamic library required for the DDK to use the CPU for model inference
ai_ddk_lib\include\HiAiMod	C++ API header file provided by the DDK for

Directory	Description
elManagerService.h	external systems
ai_ddk_lib\include\HiAiModelManagerType.h	C++ API header file provided by the DDK for external systems
ai_ddk_lib\include\HiAiAippPara.h	(Optional) C++ AIPP API header file provided by the DDK for external systems
ai_ddk_lib\include\hiai_types.h	C++ header file provided by the DDK for external systems
ai_ddk_lib\include\native_handle.h	Dependent header file of the DDK

- Model building: dependent libraries and related header file

Directory	Description
ai_ddk_lib\lib64\libhiai_ir.so	Library on which IR operator definition and graph building depends
ai_ddk_lib\lib64\libhiai_ir_build.so	Library on which IR model building depends
ai_ddk_lib\include\hiai_ir_build.h	Header file for DDK IR APIs of model building, operator definition, and model building

2.4 Directory document

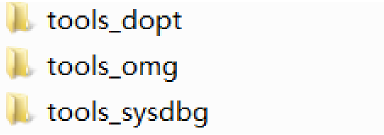
This folder contains the following documentation.

Documentation	Description
Huawei HiAI DDK V320 Release Notes	Describes the version changes and feature updates in HiAI DDK V320.
Huawei HiAI DDK V320 FAQs	Describes FAQs related to the HiAI DDK.
Huawei HiAI DDK V320 IR Model Building Instructions	Describes the model building methods and APIs. Operators under frameworks other than TensorFlow and Caffe are also supported.
Huawei HiAI DDK V320 Quick Start	Describes the HiAI DDK.

Documentation	Description
Huawei HiAI DDK V320 Model Inference and Integration Instructions	Describes the model integration and compilation methods and integration APIs.
Huawei HiAI DDK V320 Operator Specifications	Describes the restrictions of operators supported by HiAI DDK V320.
Huawei HiAI DDK V320 Acronyms and Abbreviations	Describes the acronyms, abbreviations, and terms in the HiAI DDK.
tools\tools_dopt\Huawei HiAI DDK V320 Lightweight Tool Instructions	Lightweight tool instructions
tools\tools_omg\Huawei HiAI DDK V320 OMG Tool Instructions	OMG instructions
tools\tools_sysdbg\Huawei HiAI DDK V320 System Debug Tool Instructions	Instructions of system debug tool

2.5 Directory tools

The **tools** directory contains the following directories.



tools_dopt
tools_omg
tools_sysdbg

- **tools_dopt** provides the instructions and demos of the lightweight tool.
- **tools_omg** provides the instructions of the Caffe/TensorFlow-based OMG tool and offline model building demos.
- **tools_sysdbg** is a system debug tool for the Android platform.

Their directory structures are described as follows.

Table 2-1 Directory tools_dopt

Directory	Description
tools\tools_dopt\caffe	.so files and source code used for Caffe retraining
tools\tools_dopt\tensorflow	.so files used for TensorFlow retraining
tools\tools_dopt\dopt_trans_tools	Tool for model conversion after model retraining
tools\tools_dopt\demo	Caffe and TensorFlow sample models
tools\tools_dopt\config	Configuration script of the used framework, for example, the path of Caffe code.
tools\tools_dopt\env	Docker environment configuration file of the lightweight tool

Table 2-2 Directory tools_omg

Directory	Description
tools\tools_omg\omg	Offline model generator (OMG)
tools\tools_omg\v300	V300 capability package used by the V300 OMG
tools\tools_omg\v310	V310 capability package used by the V310 OMG
tools\tools_omg\v320	V320 capability package used by the V320 OMG
tools\tools_omg\IR	IR capability package used by the IR OMG
tools\tools_omg\sample	Sample models and configuration files of model conversion with AIPP and 8-bit quantization

Table 2-3 Directory tools_sysdbg

Directory	Description
tools\tools_sysdbg\data_proc_tool	Performance data processing tool (generating *.csv files)
tools\tools_sysdbg\model_run_tool	Performance data generation tool

Directory	Description
tools\tools_sysdbg\*.so	Library on which the tool depends

3 Version Mapping

DDK Version	Typical Device Model	Kirin SoC	HiAI Version	Number of Supported Operators
V150	P20 P20 Pro Mate RS Honor 10	Kirin 970	-	90
	Nova 3 Honor play Honor Note10			
V200	Mate20 Mate20 Pro	Kirin 980	-	150
V300	Nova 5 Nova 5z Nova 5i pro Honor 9X Honor 20s	Kirin 810	100.300.xxx.xxx	178
V310	Mate 30	Kirin 990	100.310.xxx.xxx	223
V320	P40	Kirin 990 Kirin 820 Kirin 985	100.320.xxx.xxx	306

4 Integration Procedure

The integration procedure provides guidance for converting a source model to an offline model using the OMG, integrate the model inference to generate an APK, and run the APK on the Kirin SoC to accelerate the neural network.

4.1 Environment Preparations

- Use Ubuntu 16.04, or macOS to install Android Studio.

  Android Studio download address:
  https://developer.android.com/studio/index.html
- Build native code with NDK R14b or later. Alternatively, you can use CMake to compile native code.

  NDK download address:
  https://developer.android.com/ndk/downloads/index.html
- Run the OMG in **tools_omg** on Ubuntu 16.04 (64-bit).

  Download address of Linux images:
- Prepare a trained Caffe or TensorFlow model.
- Prepare a device powered by the Kirin SoC for testing the app. For details about their version mappings, see 3 "Version Mapping."

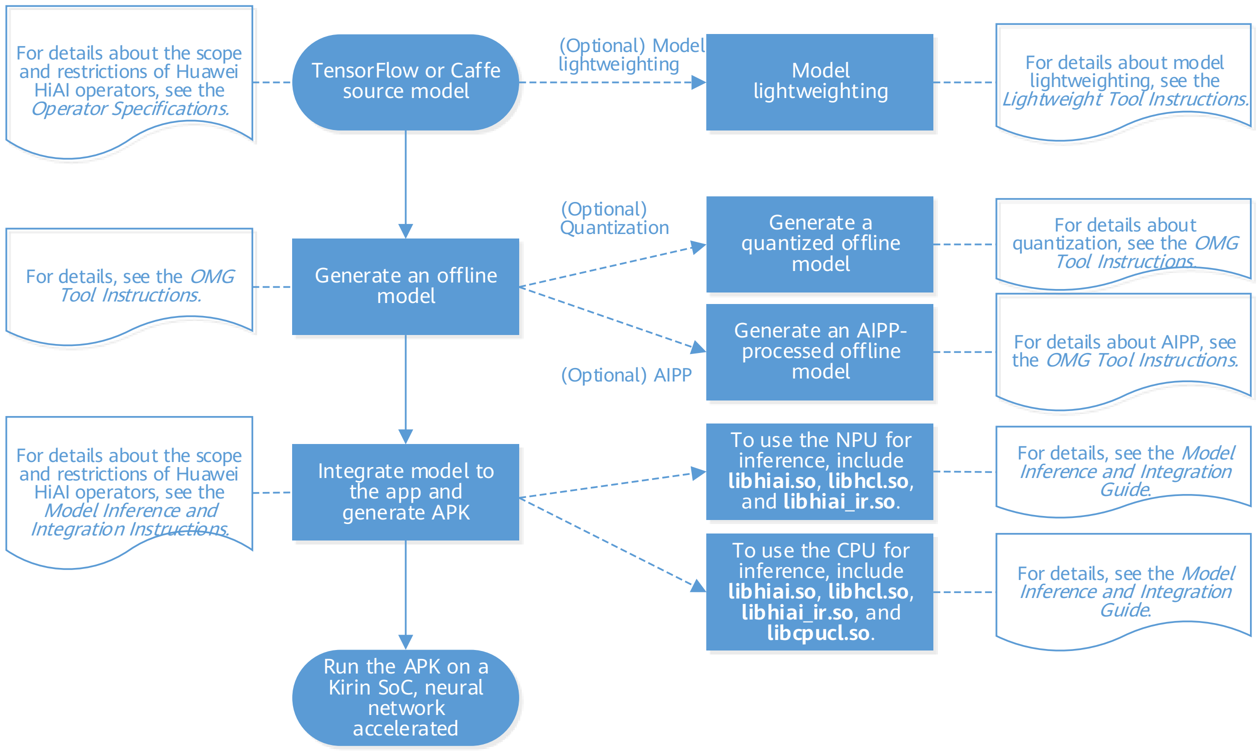
4.2 Procedure Description

Figure 4-1 shows the procedure for integrating HiAI DDK V320 to the app. Artificial intelligence pre-processing (AIPP) and quantization are optional.

NOTE

The reference documents in the following figure are for reference only.

Figure 4-1 NPU/CPU Integration procedure


For details about the scope and restrictions of Huawei HiAI operators, see the *Operator Specifications*.
TensorFlow or Caffe source model
(Optional) Model lightweighting
Model lightweighting
For details about model lightweighting, see the *Lightweight Tool Instructions*.
For details, see the *OMG Tool Instructions*.
Generate an offline model
(Optional) Quantization
Generate a quantized offline model
For details about quantization, see the *OMG Tool Instructions*.
(Optional) AIPP
Generate an AIPP-processed offline model
For details about AIPP, see the *OMG Tool Instructions*.
For details about the scope and restrictions of Huawei HiAI operators, see the *Model Inference and Integration Instructions*.
Integrate model to the app and generate APK
To use the NPU for inference, include **libhiai.so**, **libhcl.so**, and **libhiai_ir.so**.
For details, see the *Model Inference and Integration Guide*.
To use the CPU for inference, include **libhiai.so**, **libhcl.so**, **libhiai_ir.so**, and **libcpucl.so**.
For details, see the *Model Inference and Integration Guide*.
Run the APK on a Kirin SoC, neural network accelerated

Lightweight Source Model

Lightweight models can be generated by deeply optimizing source models in frameworks such as TensorFlow and Caffe, reducing the model size and accelerating the model inference speed. Currently, the non-training mode and retraining mode are supported. For details about lightweight operations, see the *Huawei HiAI DDK V320 Lightweight Tool Instructions*.

Offline Model Conversion

Caffe or TensorFlow models have to be converted into the model formats supported by the HiAI platform, and the converted offline models go through AIPP and quantization operations as required. Their application scenarios and methods are as follows:

- AIPP

  AIPP is used to preprocess images on hardware into required formats of the inference computing platform, including resizing, color gamut or image format conversion, and image pixel adjustment by subtracting the average value or multiplying a coefficient. Adaptation is implemented simply by configuring AIPP parameters or calling AIPP APIs at the software layer. In addition, AIPP improves the inference performance because it is dedicated to hardware. For details, see the AIPP model conversion and configuration instructions in the *Huawei HiAI DDK V320 OMG Tool Instructions*.

- Quantization

  Quantization converts a fp32 model into a low-bit model to save network storage space, reduce the transmission delay, and improve the computation efficiency. For details about quantization, see the related instructions in the *Huawei HiAI DDK V320 OMG Tool Instructions*.

App integration

The app integration process includes model preprocessing, model loading, model running, and model postprocessing.

- In the NPU scenario, include **libhiai.so**, **libhcl.so**, and **libhiai_ir.so** during model preprocessing. After the APK is built, the app can perform inference on the NPU. For details, see the *Huawei HiAI DDK V320 Model Inference Integration Guide*.
- In the CPU scenario, include **libhiai.so**, **libhcl.so**, **libhiai_ir.so**, and **libcpucl.so** during model preprocessing. After the APK is built, the app can perform inference on the CPU. For details, see the *Huawei HiAI DDK V320 Model Inference Integration Guide*.

5 Supported Operators

For details, see the *Huawei HiAI V320 DDK Operator Specifications*.

6 DDK Data Security

6.1 DDK Operating Mode

The mobile DDK needs to be loaded to an app during app packaging. The DDK is loaded with the app startup, and closed with the app closure without extra actions performed in the background.

6.2 DDK Permission

The DDK does not involve permission application.

6.3 DDK Data collection

The DDK does not collect any data and only accepts the data transferred by apps.

6.4 DDK Data Security Protection

The data received by the DDK is processed only on the device side and does not need to be reported to the server.