

OpenNCC CDK

软件开发手册

历史版本

| 版本 | 日期 | 修改 | 变更摘要 |
|-------|----------|----|------|
| 1.0.0 | 2020/5/7 | 王洋 | 初始版本 |
| | | | |

1. 介绍

1.1 概述

文档介绍了 OpenNCC CDK 和 OpenVINO 的关系，及使用 OpenNCC CDK 开发部署独立模式、与 OpenVINO 混合模式的方法和 Demo 程序。

1.2 支持平台

1.2.1 OpenNCC CDK 支持以下硬件和环境

- OpenNCC DK R1、OpenNCC C R1
- 主机环境：Ubuntu16.04、Ubuntu18.04、树莓派、Arm Linux(需提供工具链交叉编译)
- 支持语言：C/C++、Python3.5、Python3.7

1.2.2 OpenNCC CDK 支持的 OpenVINO 版本

OpenVINO_2019.1.144 版本，暂不支持 2019_R2 和 2019_R3*，及其后续的版本

1.3 CDK 开发包目录结构

| 目录 | 内容概要 |
|---|---|
| ncc_cdk/Docs | OpenNCC 离线文档集合 |
| ncc_cdk/NCC_View/Linux | Linux 版本的 OpenNCC View 可执行文件 |
| ncc_cdk/Public/Firmwares | OpenNCC 适配的固件目标文件 |
| ncc_cdk/Public/Library/For_C&C++/Linux | Linux 系统下 C/C++ OpenNCC CDK 静态库和 VPU 的 USB 启动引导程序 |
| ncc_cdk/Public/Library/For_C&C++/Windows | Windows 系统下 C/C++ OpenNCC CDK 静态库和 VPU 的 USB 启动引导程序 |
| ncc_cdk/Public/Library/For_Python | Python 版本 OpenNCC CDK 包，及演示程序 |
| ncc_cdk/Public/Pretrained_models/Intel_models | Intel 免费模型集合 |
| ncc_cdk/Samples/How_to/Capture video | 示例程序，使用 OpenNCC CDK 库获取视频流 |

| | |
|---|--|
| ncc_cdk/Samples/How_to/load a model | 示例程序，使用 OpenNCC CDK 库下载一个 Blob 格式的深度学习模型 |
| ncc_cdk/Samples/How_to/work with OpenVINO | 示例程序，使用 OpenNCC CDK 库让 OpenNCC 相机集成兼容 OpenVINO |
| ncc_cdk/Tools/myriad_compiler | Blob 格式文件转换工具 |
| ncc_cdk/Tools/deployment | OpenNCC 部署脚本 |

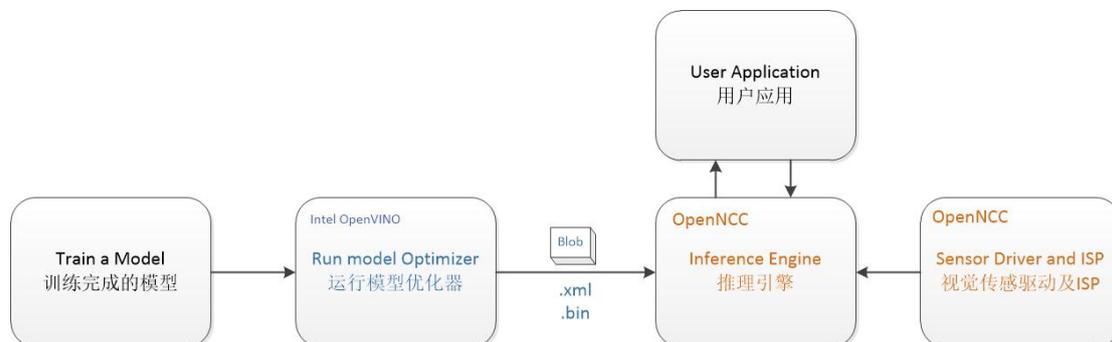
1.4 用户支持中心

请访问 <https://www.openncc.com.cn> 获取更多动态和文档

2. OpenVINO 的安装和使用

如下图，在端侧部署一个深度学习模型，需要将一个训练完成的模型经过针对 VPU 特性的模型优化和转换，以达到较高的运行性能。OpenNCC 兼容 OpenVINO 的工具集和模型格式，需要依赖 Intel OpenVINO 的模型优化器来完成模型优化和转换成 Blob 格式。使用 OpenNCC CDK 时需要安装 OpenVINO 的两种情况如下：

- 如果需要自行转换训练好的模型，那么需要安装 OpenVINO，来运行模型优化器。
- 当 OpenVINO 运行在与 OpenVINO 推理引擎的混合模式时，也需要 OpenVINO 支持。



2.1 OpenVINO 的下载

下载链接地址：

<https://software.intel.com/en-us/openvino-toolkit/choose-download/free-download-windows>，请根据自己操作系统环境选择版本。根据网速情况一般下载时间在 10 分钟，如果下载 OpenVINO 遇到问题，可以联系 Intel 或 [OpenNCC](#) 获取支持。

2.2 OpenVINO 的安装

安装指南：

https://docs.openvino-toolkit.org/latest/docs/install_guides/movidius_setup_guide.html

2.3 Intel Free 模型下载

OpenNCC 支持 OpenVINO 下生产的模型，Intel 有大量免费训练好的模型供学习参考和评测。当我们安装完成 OpenVINO 后，可以使用 Intel 下载工具下载模型集合。模型下载工具路径：`openvino/deployment_tools/tools/model_downloader/downloader.py`，常用命令如下：

- 查看全部可下载的模型：`./downloader.py --print`
- 下载指定的模型：`./downloader.py --name *`

2.4 模型的优化和格式转换

当我们需要将一个训练好模型部署到 OpenNCC 时，需要对模型进行优化和转换。安装完成 OpenVINO 后，可通过模型优化工具 `openvino/deployment_tools/model_optimizer/mo.py` 进行模型优化，具体文档见 Intel 官方文档：[Model Optimizer Developer Guide](#)。

优化完成模型后，需要进行模型转换到 Blob 格式，才能在 OpenNCC 上进行部署。在 OpenVINO 安装目录 `openvino/deployment_tools/inference_engine/lib/intel64` 下的 `myriad_compile` 工具，使用方法如下：

```
命令行终端下输入：myriad_compile -m input_xxx-fp16.xml -o  
output_xxx.blob -VPU_PLATFORM VPU_2480  
-VPU_NUMBER_OF_SHAVES N -VPU_NUMBER_OF_CMX_SLICES M (1<=N<=8,  
1<=M<=8, 且 N<=M)
```

完成格式转换后，可在 OpenNCC 上部署模型，方可参考：[ncc_cdk/Samples/How_to/load a model](#)，或者使用 OpenNCC View 界面程序添加模型来部署测试，OpenNCC View 下部署模型详见 "OpenNCC CDK 入门开始"。

2.4.1 TensorFlow 转换模型案例

以 `ssd_mobilenet_v2_coco` 为例，演示如何优化和转换一个可以在 OpenNCC 上运行的模型：

- 模型下载：
`openvino/deployment_tools/tools/model_downloader/downloader.py`
`--name ssd_mobilenet_v2_coco`，如果不能下载，可以到 https://docs.openvino toolkit.org/2019_R1.1/docs_MO_DG_prepare_model_convert_model_Convert_Model_From_TensorFlow.html 下载。
- 模型优化：`openvino/deployment_tools/model_optimizer/mo.py`
`--framework=tf --data_type=FP16 --reverse_input_channels`
`--input_shape=[1,300,300,3] --input=image_tensor`
`--tensorflow_use_custom_operations_config=/opt/intel/openvino_2019.1.`

```
144/deployment_tools/model_optimizer/extensions/front/tf/ssd_v2_support.json --tensorflow_object_detection_api_pipeline_config=pipeline.config --output=detection_classes,detection_scores,detection_boxes,num_detections --input_model=frozen_inference_graph.pb。
```

- 模型编译:
`openvino/deployment_tools/inference_engine/lib/intel64/myriad_compile -m frozen_inference_graph.xml -o frozen_inference_graph.blob -VPU_PLATFORM VPU_2480 -VPU_NUMBER_OF_SHAVES 8 -VPU_NUMBER_OF_CMX_SLICES 8`

获取到 `frozen_inference_graph.blob` 文件，此文件可以下载到 OpenNCC 运行。参考例子：`ncc_cdk/Samples/How_to/load a model`

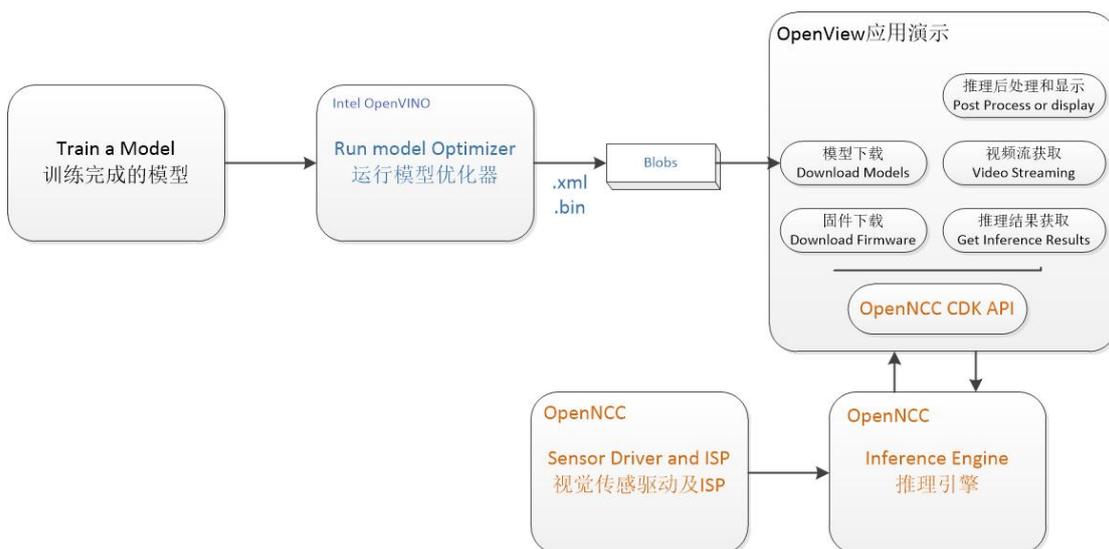
2.4.2 Caffe 转换模型案例

Caffe 模型的优化与转换，与 TensorFlow 模型过程唯一区别是，模型优化过程差异，需要输入以下命名：

```
$openvino/deployment_tools/model_optimizer/mo_caffe.py --input_model <INPUT_MODEL>.caffemodel --data_type=FP16
```

3. OpenNCC 独立模式开发

独立模式是 OpenNCC 的默认工作模式，OpenNCC 相机在端侧完成 ISP 处理，并在相机上运行推理引擎，将推理结果反馈给主机端。OpenView 是工作于独立模式的典型应用，如下图所示：



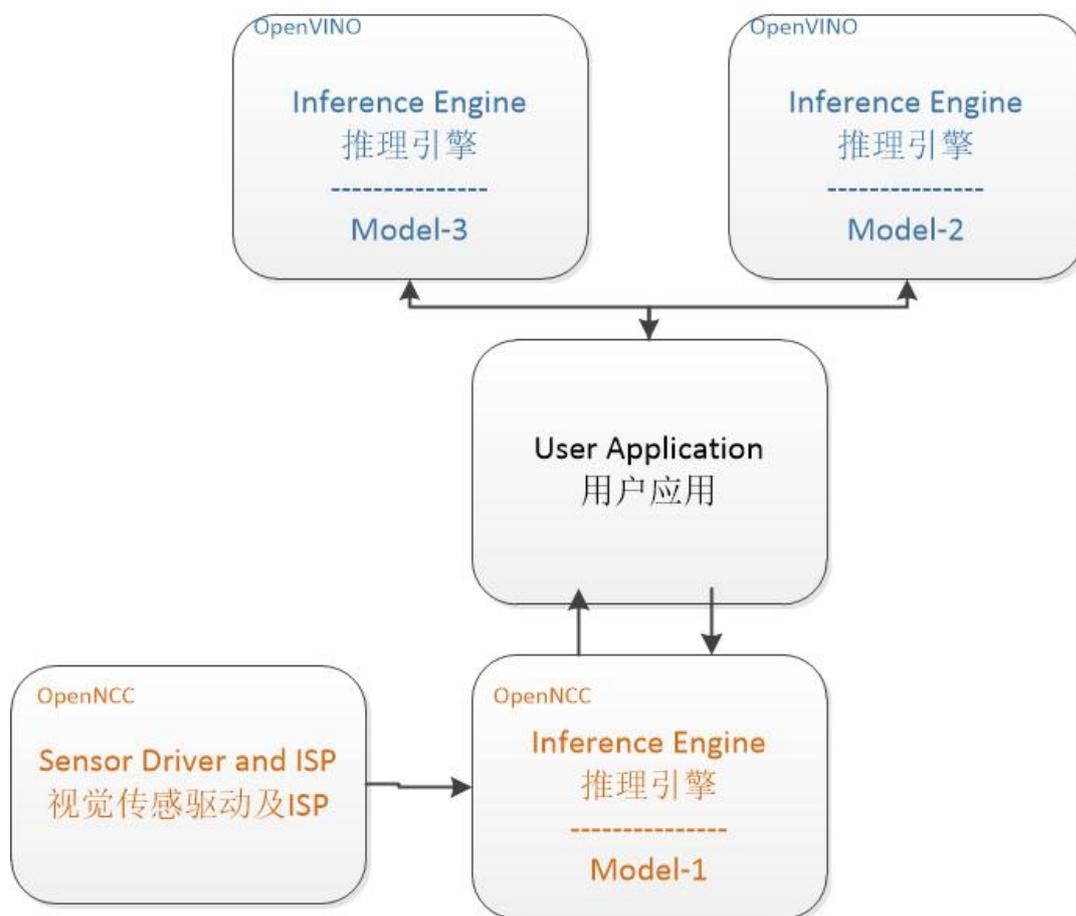
示例程序：`ncc_cdk/Samples/How_to/load a model` 也很好展示了 OpenNCC 工作需要全部流程。由于不同的深度模型有差异化的推理输出结果，OpenNCC CDK 对不同格式结果支

持在不断增加中，如果用户无法在 CDK 下找到合适的后处理解析模型，需要自己参考 `ncc_cdk/Samples/How_to/load a model` 并结合自己应用场景来编写后处理代码

4. OpenNCC 混合模式开发

当需要解决一些复杂应用场景，需要多个网络模型组合处理、OpenNCC 端侧计算性能无法满足、或者端侧处理完成后需要到边缘侧集中后处理时，往往需要进行系统扩增。将实时性诉求高的模型运行在 OpenNCC 端侧，其他模型运行在后处理边缘机或云端。

如图，Model-1 运行在 OpenNCC 端侧，完成对视频流的前处理。OpenNCC 将一级处理模型结果返回用户应用程序，Model-1 和 Model-2 完全运行于 OpenVINO 推理引擎下，实现后续处理。



`ncc_cdk/Samples/How_to/work with OpenVINO` 演示了如何让 OpenNCC 和 Host PC 上 OpenVINO 组合实现一个分布式 AI 系统。