

OpenNCC CDK

入门文档

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

此文档介绍 OpenNCC 部署基本概念，和 OpenNCC 的独立、协处理两种工作模式，展示使用 OpenNCC DK 套件下的 OpenNCC View 快速完成应用部署演示。

1.2 支持的平台

OpenNCC CDK 支持以下硬件和环境

- OpenNCC DK 套件 R1
- 主机环境：Ubuntu16.04、Ubuntu18.04、树莓派、Arm Linux(需提供工具链交叉编译)

OpenNCC View 支持的环境

- Ubuntu16.04,Ubuntu18.04
- QT 5.9.9 及以上

1.3 用户支持中心

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

2. OpenNCC 运行机制

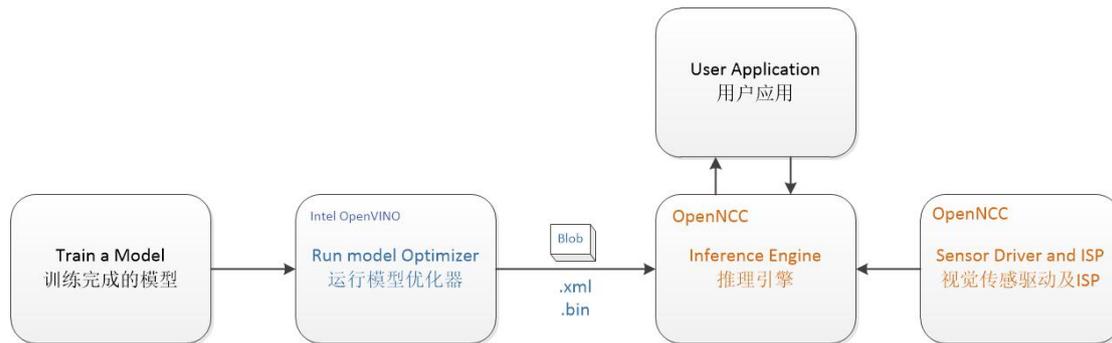
从一个模型训练环境到嵌入式部署，是一个非常重任务的工作，需要对深度学习的框架掌握，如常用的：Caffe*, TensorFlow*, MXNet*, Kaldi* 等。此外掌握部署的嵌入式平台非常重要，需要了解平台性能，系统架构特点，结合平台特点需要对训练的模型框架进行优化，并最后调优移植部署到嵌入式平台。OpenNCC 专注于深度学习模型快速部署，兼容 Intel OpenVINO，并针对嵌入式图形图像应用场景，在端侧完成了从 2MP 到 20MP 不同分比率传感器集成，端侧实现了可部署专业级别的 ISP，可将 OpenVINO 优化转换后模型文件动态下载到端侧 OpenNCC 相机，实现深度学习模型的快速部署。

同时 OpenNCC 设计了独立工作模式、混合工作模式和协处理计算棒模式来适配不同的工作应用场景。本文档重点介绍，使用 OpenView 让 OpenNCC 快速实现独立工作模式，及安装 OpenVINO 后，让 OpenNCC 运行在协处理计算棒模式。

2.1 OpenNCC 独立模式及 OpenView

独立模式下，OpenNCC 独立运行一个深度学习模型，并将推理结果通过 OpenNCC CDK API 反馈给用户。

应用程序部署流程如下图：



- 按照 OpenVINO 文档，为特定的训练框架配置模型优化器(Configure Model Optimizer)
- 运行模型优化器(Model Optimizer)，基于训练好的网络拓扑、权值和偏差值等可选参数产生一个优化后的 IR 文件，然后再运行 myriad_compile 将 IR 文件生成 BLOB 文件。详细步骤参考文档“OpenNCC CDK 软件开发手册”。
- 在应用程序上，集成使用 OpenNCC CDK 下载优化完成后的 BLOB 模型文件，见 CDK 下 Samples/How_to/Load a model 的演示程序。

OpenView 是集成了 OpenNCC CDK 的带操作界面的应用演示程序，也可以使用 OpenView 来部署模型，获取测试结果。

2.1.1 OpenView 安装

使用 OpenView 来实现 OpenNCC 独立工作模式，需要先完成安装步骤：从 www.openncc.com.cn 获取 CDK 开发包。

Ubuntu 下运行 OpenNCC View

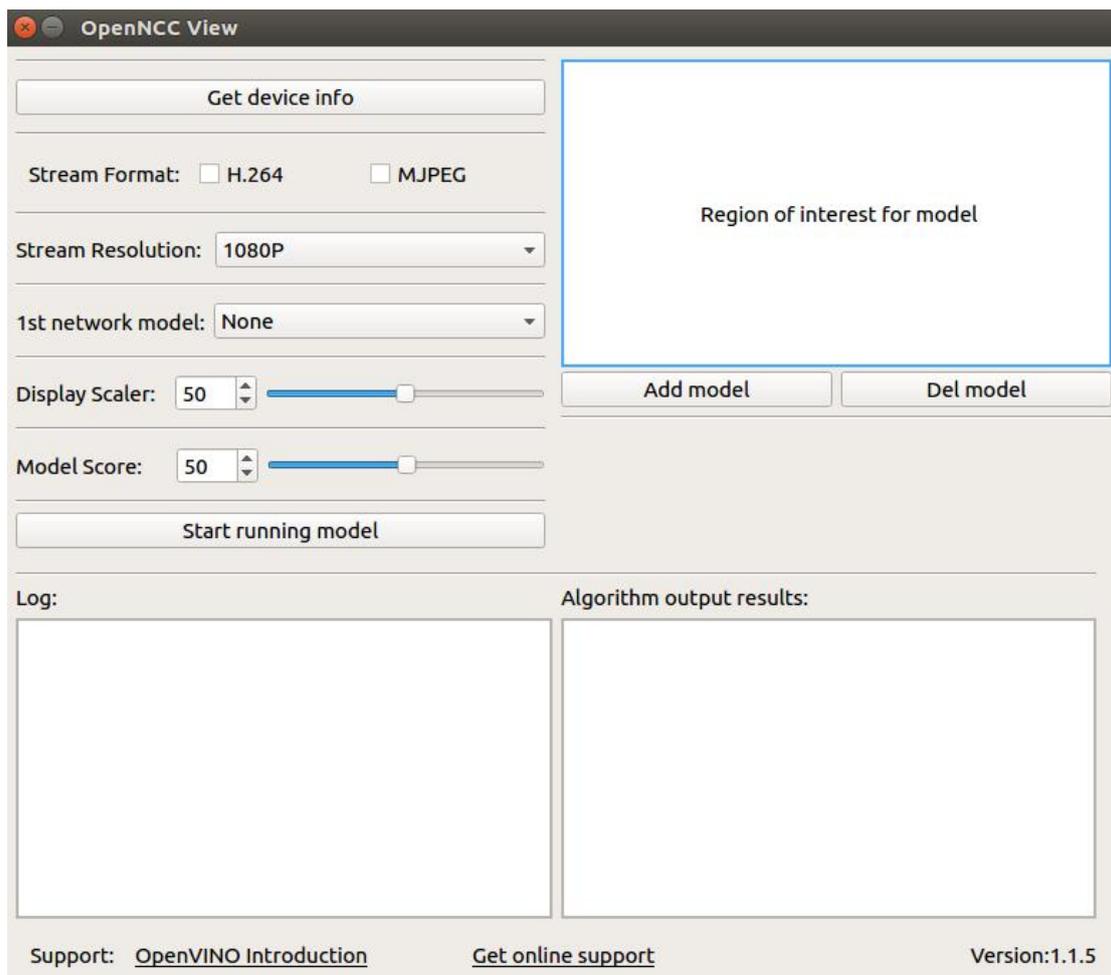
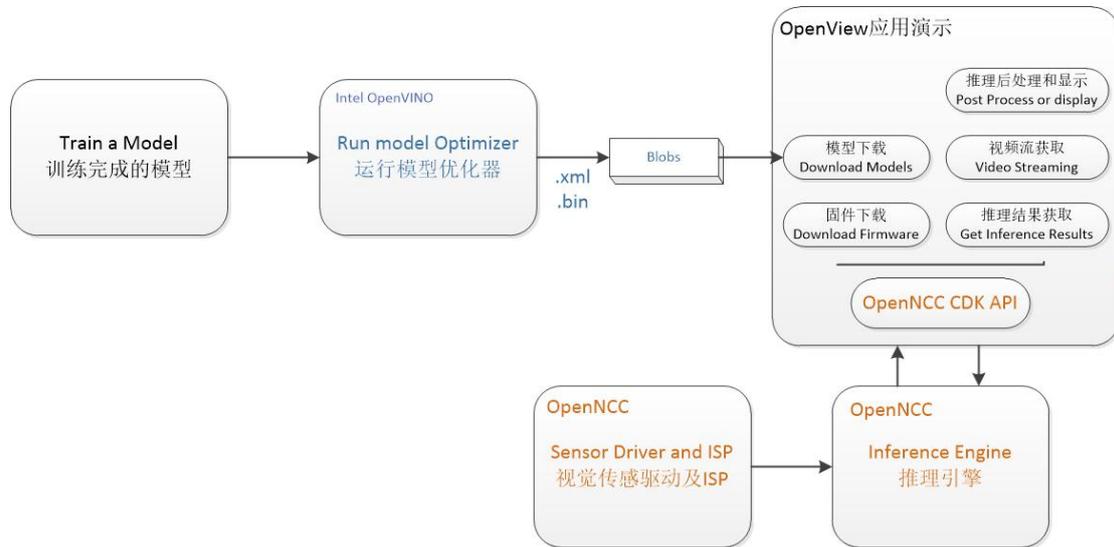
运行 OpenNCC 的 CDK 包下 Tools/deployment 目录下 install_NCC_udev_rules.sh 脚本，在命令终端输入：sudo ./install_NCC_udev_rules.sh，以获取 OpenNCC 相机自动挂载权限，重启电脑。

进入 CDK 下的 OpenNCC 的目录，终端输入命令：\$./OpenNCC 打开 NCC 软件界面（若运行失败，可尝试 sudo ./OpenNCC）

2.1.2 OpenNCC View 使用说明

OpenNCC View 是一款用于快速体验 OpenNCC DK 开发套件的软件，运行 OpenNCC View 下默认模型不依赖 OpenVINO，NCC View 集成了 NCC CDK 全部 API,可以完成 OpenNCC DK 在独立模式下实现对相机的连接,固件和 AI 模型的下载，及完成视频流显示与算法结果的后处理。用户可以通过友好的界面，来操作和控制相机。

OpenNCC View 在内测中，未来会开放源码，便于用户进行二次开发，欢迎申请源码参与内测，帮助优化功能和体验。



Get Device Info: 软件初始状态展示的功能为基础功能，隐藏功能需要用户先点击 Get Device Info，获取设备与电脑 usb 连接信息及 NCC 设备模块信息，若连接的设备支持，能够解锁更多功能

选项。（例如 NCC 与电脑通过 **usb 3.0** 连接，可解锁 **yuv** 出流显示视频；NCC 装配了 4K 模组，可解锁 4K 分辨率显示）

Stream Format: 选择 NCC 相机 USB 视频输出格式，目前支持 YUV420P, H.264, MJPEG 格式。（选择 YUV420P 前需先点击 **Get Device Info**，仅在 **usb 3.0** 连接时可选）

Stream Resolution: 更换 NCC 相机的视频分辨率，可选 1080P 或 4K. 4K 分辨率支持需要选装相应 4K 摄像模块。（选择 4K 前需先 **Get Device Info**，仅在模组支持 4K 时可选）

1st network model: 选择算法模型，选择 **None** 即不加载模型，仅出流显示视频，而选择加载算法模型后，可以通过框选 **ROI** 区域，仅对区域内的场景进行识别。

Model Score: 加载算法后，对物品的识别计算结果是有分数的，当超过了某一分数阈值，才会在视频中框选，而 **Model Score** 即为控制阈值的选项，根据用户需求，实时调整识别的最低分数（默认值为 0.5）

ROI: 配合算法模型使用，加载模型后，如果需要仅对某一区域进行算法识别，可以手动点击鼠标左键拖动，框选出自己感兴趣的区域，仅对区域内的场景进行识别

Display Scaler: 视频显示时，由于不同电脑的分辨率不同，按原本尺寸显示 1080P 或 4K 的视频，可能会出现视频大小超出桌面大小的情况，用户可通过 **Display Scaler** 实时控制视频显示窗体大小（默认值为 0.5）

Add model: 导入用户自己的模型文件，添加后即可在 **1st network model** 中选择自己的模型，但由于不同模型的算法解析方式不同，**OpenNCC View** 无法提供通用的解析器对用户自定义的算法模型进行解析，用户可参考 **OpenNCC View** 的代码，自行开发适用于自己算法的解析器

Del model: 删除用户导入的模型文件，删除后即会在 **1st network model** 中移除用户自己的模型

Start running model: 点击即开始加载固件，并根据用户的选项，显示视频窗口

Log: 显示 NCC 工作时的 **log**，用户可在此处查看设备的运行状况，以及关于设备异常状态的提示信息

Algo Results: 显示算法结果，当用户选择加载算法模型，且在实际场景中有被识别到的算法模型，即会有算法结果在此处打印（如被识别模型相对当前窗口的坐标信息，及算法计算出的识别分数等）

2.1.3 OpenView 更换模型

OpenView 支持物体检测模型(**Object Detection Models**)的更换，便于用户免编程部署更换自己训练的物体检测模型。

2.1.3.1 OpenView 默认支持的演示模型

OpenView 默认使用的预训练模型来自于 **OpenVINO** 下的 [Free Open model zoo](#).

- [face-detection-adas-0001](#)
- [face-detection-retail-0004](#)
- [face-person-detection-retail-0002](#)
- [person-detection-retail-0013](#)
- [pedestrian-detection-adas-0002](#)
- [pedestrian-and-vehicle-detector-adas-0001](#)
- [vehicle-detection-adas-0002](#)
- [person-vehicle-bike-detection-crossroad-0078](#)
- [vehicle-license-plate-detection-barrier-0106](#)
- [classification](#)

每个模型的详细信息，可点击模型列表的链接获取。

2.1.3.2 推理后处理解析

模型在 OpenNCC 完成推演后，通过 OpenNCC CDK API 获取实时的推演结果，OpenView 针对物体检测这类模型实现了推演结果的通用解析器。

OpenView 的推演后处理支持如下格式输出：

输出的数据形态：[1, 1, N, 7]

N 是当前帧下检测到的物体框数量

对于每个检测框，包含以下信息格式：

[image_id, label, conf, x_min, y_min, x_max, y_max]，其中：

image_id - ID of the image in the batch

label - predicted class ID

conf - confidence for the predicted class

(x_min, y_min) - coordinates of the top left bounding box corner

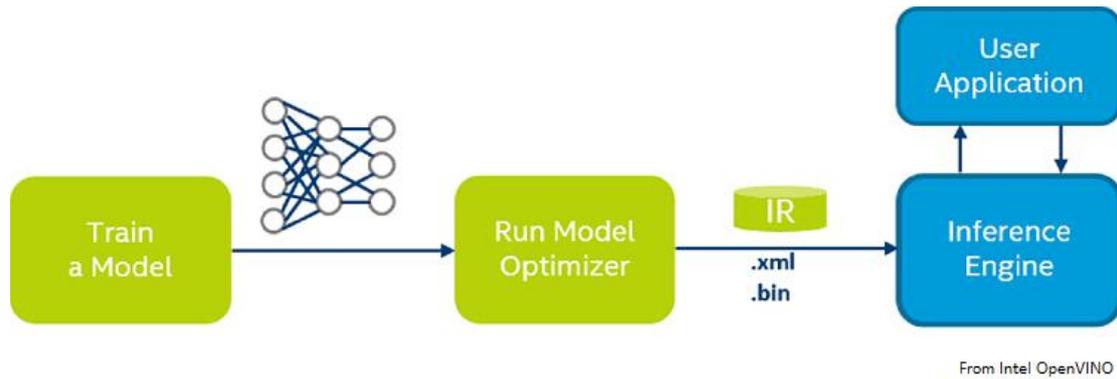
(x_max, y_max) - coordinates of the bottom right bounding box corner.

用户可以训练自己模型，并将输出层按照以上格式定义，可以免编程使用 OpenView 的模型导入功能添加并测试物体检测模型。如果用户需要添加其他输出格式的模型，参考文档 OpenNCC CDK 软件开发手册，编码实现。

2.2 协处理计算棒模式

OpenNCC 的协处理模式，类似与 Intel NCS2 计算棒。这种工作模式下，OpenNCC 的视觉传感器不工作，用户可以单独使用 OpenNCC C 来实现完全兼容 OpenVINO 环境。

OpenVINO 典型的深度学习模型部署流程如下：



- 按照 OpenVINO 文档，为特定的训练框架配置模型优化器(Configure Model Optimizer)
- 运行模型优化器(Model Optimizer)产生一个优化后的 IR 文件，基于训练好的网络拓扑、权值和偏差值等可选参数。
- 将优化生成的 IR 文件下载到 OpenNCC 上运行推理引擎(Inference Engine)，具体参考 OpenVINO 文档：[Inference Engine validation application](#) 和 [sample applications](#)。
- 将 Public/Firmwares/MvNCAPI-ma2480.mvcmd 复制并且替换 openvino 安装目录下的 openvino/inference_engine/lib/intel64/MvNCAPI-ma2480.mvcmd。(备注：替换前必须备份 MvNCAPI-ma2480.mvcmd，使用 NCS2 推理时需要恢复该文件)

2.3 独立模式和协处理模式区别

如下图右侧是 OpenNCC 的独立模式，左侧是 OpenNCC 的协处理模式(类同 Intel NCS2)。

当我们需要部署一个基于视觉的深度学习模型时，首先我们需要获取一个高质量的视频流，然后运推理引擎来把输入的图像数据进行计算，最后输出结果。左侧的协处理模式，我们需要一个 OpenNCC C 或者 Intel NCS2 实现端侧推理，同时我们需要从一个摄像机获取视频流，并将视频帧通过 USB 发送给 OpenNCC C。而右侧的独立模式，不需要额外的摄像机来获取视频流，我们只需要将模型下载到 OpenNCC 后，就可以获取到推演结果。

