Smart sense

# 项目需求

## 1 需求

1. 编写SDK完成相应功能的SDK发布
2. 负责分割算法开发和维护
   1. 当前分割模型分割区域不准确，无法提供准确的分割区域

## 2 进度计划

### 2.1 SDK进度

1. 进度表

|  |  |  |  |
| --- | --- | --- | --- |
| SDK版本 | 交付日期 | 功能接口 | 新增功能 |
| V0.0 | 0619 | 详见 smartSenceSDK.docx |  |
| V1.0 | 0705 | 详见 smartSenceSDK.docx |  |
| V1.1 | 0825 | 详见 smartSenceSDK.docx | 1. 分割模型输出带颜色的mask 2. 分辨率720P以上 |

1. 效果展示

|  |  |  |  |
| --- | --- | --- | --- |
| L_4292 | result_chuanglian | resultChair | resultWall |
| L_082 | resized_dep | resized_seg | resized_seg_wall |
|  | resized_seg_rect |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 分辨率 | 模型 | 推理速度 | 平台（算力） |
|  | 深度估计模型 | 15.7790ms | 联咏nt98530(2T) |
|  | 分割模型 | 34.7500ms | 联咏nt98530(2T) |

|  |  |  |
| --- | --- | --- |
| Sdk并入耗时 | 本地测试 | 软件测试 |
|  | 51.10ms | 71.886ms |

### 2.2 分割算法进度

# SDK开发记录

## 0 工具使用

### 0.1 df -h 查看磁盘空间

### 0.2 取消乱码显示 dmesg -n1

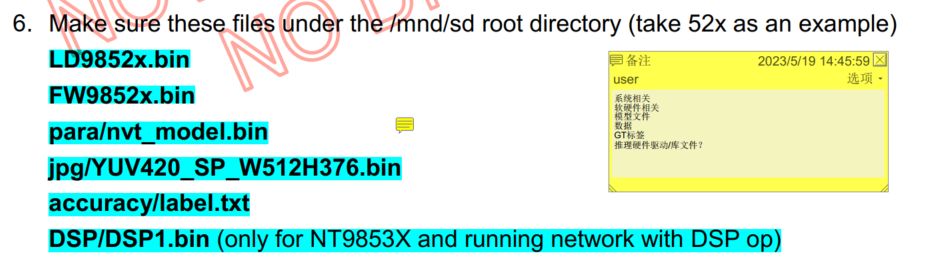
## SDK开发

### 1.1 novatek文档阅读总结

1. 目标：
   1. 熟悉硬件性能
   2. 获取模型移植的方案（用什么推理引擎，有无demo等）
2. 硬件性能
   1. 相关文档 NT98530\_Differences\_Between\_NT98528\_and\_NT98530\_en.pdf
      1. 系统和架构：Linux 4.19.148 ARM64
      2. 算力： CNN with computing power up to **2 TOPS**
      3. 其硬件名为DSP（类比GPU NPU？）
3. SDK和NNTool
   1. Sdk包含了不仅仅是ai相关的代码，还包含系统SDK

**Novatek AI2 SDK - Quick\_Start\_en.pdf文档阅读**

1. 针对不同的芯片型号如：NT98530有如下不同配套使用的东西
   1. DLA Engine 深度学习加速器推理引擎
   2. Nvt AI tool
   3. Model bin format
      1. 模型类型为nvt\_model ？
   4. Nvt AI2 SDK
   5. DSP bin（digital signal processor）
   6. 关于nvt AI tool nvt AI2 SDK以及DSP的关系文档如下描述
      1.  Nvt AI Tool: using V01.06.2209070 or later, to **generate network** with DSP op.
      2.  Nvt AI2 SDK: using 02.16.2209020 or later, to **running** network with DSP op
2. 疑问点
   1. 文档讲的的demo中几个文件的作用

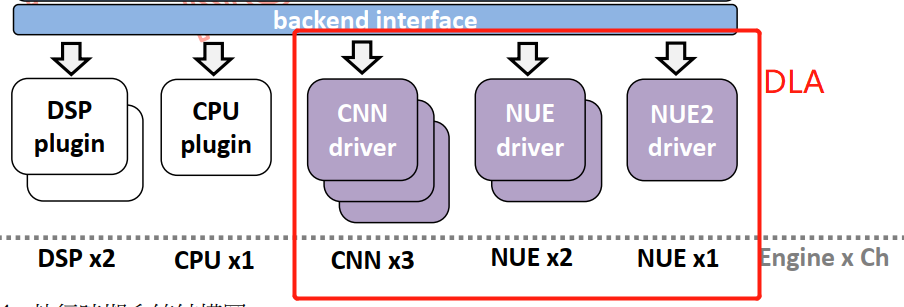


**Novatek AI2 SDK - Programing\_Guide\_en.pdf**

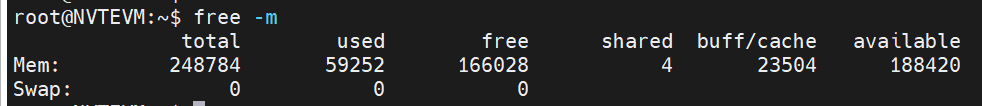
1. AI 开发文档详细介绍
   1. AI结构简介
      1. 从硬件层面本芯片的神经网络运算核心有三类 DLA CPU 和 DSP
      2. 其中，对于DLA核心，AI2支持三种推理引擎分别为，CNN NUE 以及NEU2
   2. API执行网络时，通过proc\_id来区分不同网络的运行（也就是说可以支持同时运行多个网络），proc\_id 上限为16 即， 0-15

**板子相关信息总结**

1. 硬件信息
   1. 推理引擎：DLA三种推理引擎CNN NUE NUE2



* 1. 算力：2T
  2. 内存：



### 1.2 Nt98530SDK开发框架搭建

#### 1.2.1 使用板子自带demo跑通测试板子环境

1 板子连接和文档传输相关方法记录（usb工具进行文件传输 串口进行连接做远程控制）

* 1. **nt98530板子连接**
     1. **连接串口工具到笔记本上 使用mobaxterm测试连接是否通**
     2. **连接USB工具到笔记本上**
  2. **连接后使用demo做测试**
     1. **USBCameraTool 使用usb往nt98530板子上发送下载文件的工具**
        1. **方法： 在本文文件夹导航栏cmd打开终端**
     2. **发送文件 USBCameraConsoleTool.exe -v 1ff7 -p 0525 put\_file xxx xxxx**
        1. **USBCameraConsoleTool.exe -v 1ff7 -p 0525 put\_file my\_ai\_pattern.zip /tmp/1.zip**
     3. **下载文件 USBCameraConsoleTool.exe -v 1ff7 -p 0525 get\_file xxx xxxx**

2 文件组织搭建

* 1. **按照文档组织文件结构执行命令测试文档提供的demo**
  2. **E:\Project\2023记录\smart sense\开发资料\NNTool\0308\_Novaic\_v01.08.2303060\_release\_exe\Doc\Novatek AI2 SDK - Quick\_Start\_en.pdf 1.5.1进行文件组织拷贝到板子上做运行测试即可**
  3. **取消乱显示 dmesg -n1**

#### 1.2.2 搭建SDK以demo中的模型为例测试跑通跑

1. 按照SDK中的Demo组织代码进行编译测试

## 使用SDK步骤汇总

### 模型转换

**模型转换环境在服务器 10.22.23.2上**

1. Pytorch转onnx
2. Onnx 转novatek上的onnx
   1. 按照例子修改gen\_config.txt
      1. 修改内容记录

|  |  |
| --- | --- |
|  |  |
|  |  |

* + 1. 执行python脚本实现模型转换
       1. 深度估计模型转换案例

python ../toolchain/closeprefix/bin/compiler/frontend/onnx-onnx/onnx2novaonnx\_converter.py --input ./tmpFolder/tmp/xxx\_320x192.onnx --output ./nvtai\_tool/input/model/customer/inception\_smartSense/deploy.onnx

* + - 1. 分割模型转换案例

python ../toolchain/closeprefix/bin/compiler/frontend/onnx-onnx/onnx2novaonnx\_converter.py --input ./tmpFolder/segment/bisenetv1\_ade20k.onnx --output ./nvtai\_tool/input/model/customer/smartSense\_seg/deploy.onnx

* 1. Onnx -> bin
     1. 执行可执行文件实现novatek onnx转bin模型文件
        1. 深度估计模型
           1. ../toolchain/closeprefix/bin/compiler.nvtai --config-dir ./nvtai\_tool --pattern-name 00001\_inception\_smartSense --chip 530
        2. 分割模型
           1. ../toolchain/closeprefix/bin/compiler.nvtai --config-dir ./nvtai\_tool --pattern-name smartSense\_seg --chip 530

### 2.2 Sim(仿真看量化前后差别)

1. 按照例子修改sim\_config.txt
   1. 修改点记录

|  |  |
| --- | --- |
|  |  |

### 2.3 板端推理、模型合入SDK

#### 2.3.1 板子和PC互传数据

1. PC文件往板子传 在tool路径下文件导航栏输入 cmd 打开windows终端
   1. USBCameraConsoleTool.exe -v 1ff7 -p 0525 put\_file xxx xxxx
   2. Eg：1)USBCameraConsoleTool.exe -v 1ff7 -p 0525 put\_file my\_ai\_pattern.zip /tmp/1.zip
2. 板子端往PC传数据
   1. USBCameraConsoleTool.exe -v 1ff7 -p 0525 get\_file xxx xxxx

# 分割算法开发记录

**列出当前问题 增加解决思路和过程数据 有时间做一下分割领域的调研（数据集+算法[论文]）**

## 1 当前所用算法相关连接

[GitHub - CoinCheung/BiSeNet: Add bisenetv2. My implementation of BiSeNet](https://github.com/CoinCheung/BiSeNet)

### 1.1 分割模型执行环境

经测试当前算法的执行环境在本地rknntoolkit2中刚好可满足

可以使用该数据集对其进行重新训练改善分割不准的问题

[GitHub - nightrome/cocostuff: The official homepage of the COCO-Stuff dataset.](https://github.com/nightrome/cocostuff)

## 2 重训练前后Demo效果和性能指标对比

### 2.1 原始模型Demo展示效果 （branch master）

1. 效果展示demo
   1. python tools/demo.py --config configs/bisenetv1\_ade20k.py --weight-path checkpoints/model\_final\_v1\_ade20k.pth --img-path ./L\_271.png
2. 性能评估：

### 2.2 重训练模型效果及评估指标 （branch smartSense）

1. 效果展示demo
2. 性能评估：

## 3 模型重训练

### 3.1 模型优化需求点

1. 更改分割监督类别

enum SEGCLASS{

WALL\_ = 0, //墙

FLOOR\_ = 1, //地板

CEILING\_ = 2, //天花板

WINDOW\_ = 3, //窗户

PERSION = 4，

TABLE\_ = 5, //桌子

CURTAIN\_ = 6, //窗帘

CHAIR\_ = 7, // 椅子 需要外加 类别19

};

1. 更改最终输出尺寸为64\*64 输出分辨率变大些
2. 使用自有数据集进行训练

### 3.2 模型优化步骤记录

#### 3.2.1 使用cocostuf跑通模型训练代码

1. Cocostuf下载路径
   1. https://github.com/nightrome/cocostuff

#### 3.2.1 魔改配置文件和数据加载代码实现自有数据集上的训练