

## T710 Imagination AI NDK2.2 环境简介

适用产品信息	T710
适用版本信息	Ubuntu 16.04
关键字	AI, NDK

Unisoc Confidential For hiar

## 声明

本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

## 版本历史

版本	日期	备注
V1.0	2019/10/17	初稿

Unisoc Confidential For hiar

# 目录

<b>1 前言</b>	<b>1</b>
1.1 范围	1
1.2 缩略语	1
<b>2 环境安装</b>	<b>2</b>
2.1 软件依赖	2
2.2 安装配置	2
2.2.1 配置 python 版本	2
2.2.2 配置安装目录	2
2.2.3 配置 caffe 目录	2
2.3 执行安装	2
<b>3 模型转换</b>	<b>3</b>
3.1 进入 NDK 环境	3
3.2 需要准备的模型相关文件	3
3.3 模型转换 demo 脚本	4

# 1 前言

## 1.1 范围

本文档主要介绍了 Imagination AI NDK2.2 的环境安装以及模型转换指引。

本文档的使用人员主要包括：

需要使用 T710 Imagination AI 进行 AI 应用开发的软件开发人员以及客户。

## 1.2 缩略语

名称	全称	定义
NDK	Network Development Kit	AI 网络开发工具包

Unisoc Confidential For hiar

## 2 环境安装

### 2.1 软件依赖

准备安装有 Ubuntu 16.04 操作系统的 PC，并预先安装以下软件：

python (或 python3)  
python-pip (或 python3-pip)  
git  
curl  
cmake  
virtualenv  
caffe 1.0

### 2.2 安装配置

#### 2.2.1 配置 python 版本

Imagination NDK2.2 支持 python2 和 python3，通过修改 install\_rel\_2\_2.sh 脚本中的 python\_ver 来配置 python 版本：

```
python_ver=2    //使用 python2  
python_ver=3    //使用 python3
```

#### 2.2.2 配置安装目录

修改 scripts/ndk\_vars.sh 中的 NNA\_ROOT 来配置安装目录：

```
export NNA_ROOT=${HOME}/img_nna_2_2
```

#### 2.2.3 配置 caffe 目录

修改 scripts/ndk\_vars.sh 中的 CAFFE\_ROOT 来配置 caffe 目录(caffe 请自行安装，这边只是配置已安装好的 caffe 路径供 NDK2.2 使用)：

```
export CAFFE_ROOT=${HOME}/caffe
```

### 2.3 执行安装

执行脚本 ./install\_rel\_2\_2.sh 安装 NDK2.2。

## 3 模型转换

### 3.1 进入 NDK 环境

执行模型转换前，需要先进入 NDK 环境。

假设 NDK 安装目录为 NNA\_ROOT，通过以下命令进入 NDK 环境：

```
source ${NNA_ROOT}/scripts/ndk_vars.sh。
```

### 3.2 需要准备的模型相关文件

将以下文件放置到同一文件夹：

**已经 freeze 的模型文件。**

**IO.json 和 test\_file.json 配置文件，本安装包下 json\_sample 目录分别提供了 caffe 和 tensorflow 的示例：**

IO.json 文件用于描述网络输入节点信息：

```
[
  {
    "shape": [                //shape 需要和 layout(NCHW)属性对应
      1,
      3,
      227,
      227
    ],
    "name": "data",           //input 节点名称, 对于 tensorflow , 名称需要加_IMG
    "type": "INPUT",          //input 节点 type
    "layout": "NCHW",         //layout, 统一为 NCHW , tensorflow NHWC 会先转换为 NCHW
    "dtype": "float32"        //网络输入数据类型
  }
]
```

test\_file.json 用于描述样例图片集的信息：

```
[
  {
    "image_path": "/ai_data/imagenet_picked/",
    "mean": [104.0, 117.0, 123.0],
    "extension": "JPEG",
    "raw_scale": 1.0,
```

```

        "input_scale": 1.0,
        "name": "data",          //input 节点名称, 对于 tensorflow , 名称需要加_IMG
        "grayscale": 0,
        "channel_swap": [2,1,0]
    }
]

```

test\_file.json 各字段的释义请参考 Imagination 官方文档：  
《PowerVR\_IMGNNVM\_NDKToolsSoftwareSpecification.pdf》section 3.1.5 “Test File”小节。

### 3.3 模型转换 demo 脚本

**`\${NNA\_ROOT}/scripts` 下提供了以下模型转换 demo 脚本：**

caffe\_model\_compile.sh: caffe 模型(图片分类)转换脚本  
tf\_model\_compile.sh: tensorflow 模型(图片分类)转换脚本  
onnx\_model\_compile.sh: onnx 模型(图片分类)转换脚本  
tf\_ssd\_compile.sh: tensorflow 模型(SSD)转换脚本

**各 demo 脚本都提供如下参数：**

-m: 模型文件(参考 2.2 节需准备的模型文件)所在文件夹路径，必带参数  
-b: 模型转换 bitdepth，默认值 8，目前可选 8 或者 16

**脚本使用示例：**

./caffe\_model\_compile.sh -m \${model\_path} -b 8 //转换 8bit 模型(可省略-b 参数)  
./caffe\_model\_compile.sh -m \${model\_path} -b 16 //转换 16bit 模型

**模型转换输出：**

caffe 转换 mbs 输出在：`\${NNA\_ROOT}/caffe/[与输入文件夹同名]/model.mbs.bin  
tensorflow(包括 tensorflow SSD 网络)转换 mbs 输出在：`\${NNA\_ROOT}/tf/[与输入文件夹同名]/model.mbs.bin  
onnx 转换 mbs 输出在：`\${NNA\_ROOT}/onnx/[与输入文件夹同名]/model.mbs.bin