星宸科技

AI 在线开发平台使用

版 本： V1.0

日 期： 2025-08-07

Revision

Version	Modification	Date
1.0	Initial version	20250807

Content

目录

1 概述

AI 在线开发平台是将我司开发的 **AI 模型转换工具** IPUToolChain 的 Docker 环境**部署到**到云端，方便开发者可以快速的尝试并开发 AI 的应用。

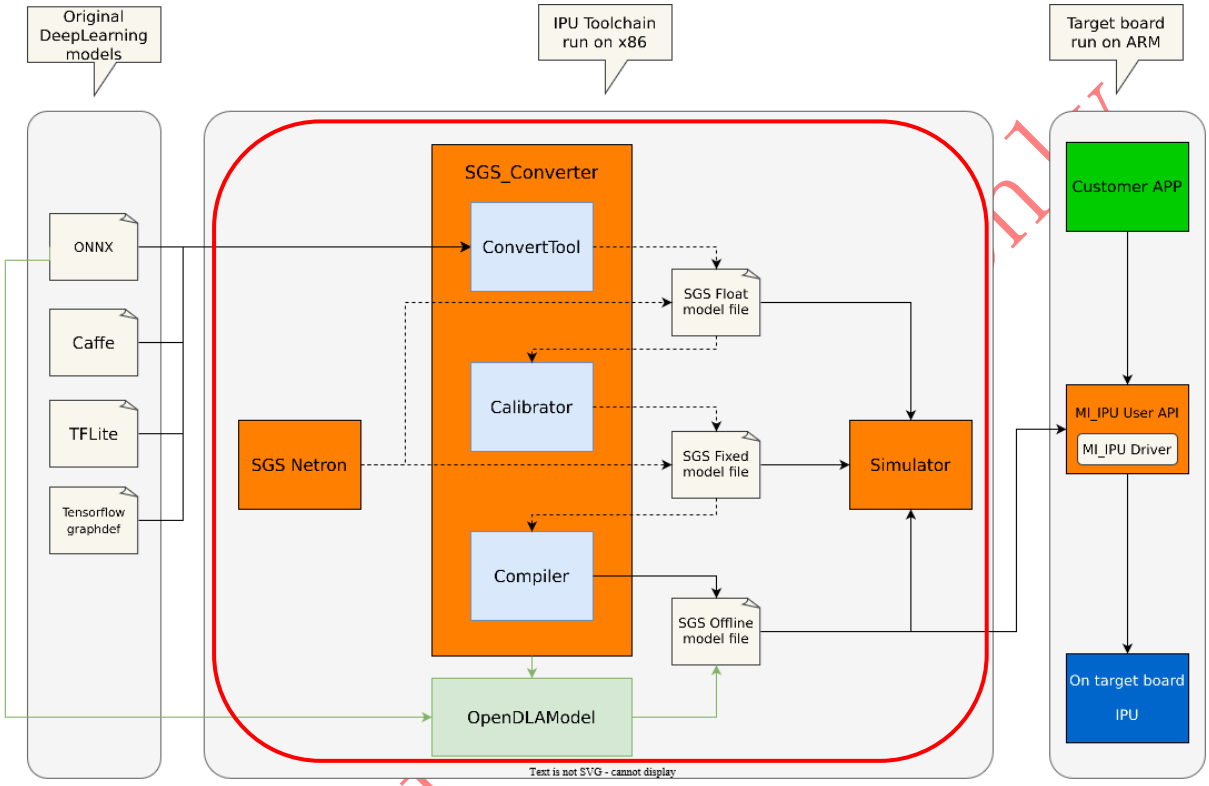


Figure 1 AI 在线开发平台整体框架

说明：

1. 上图中的 OpenDLAModel，是 Sigmstar 提供的开源模型算法的集合, 它封装了 IPU Tool Chain 的相关功能，方便客户快速的转换出离线模型。如果想开始第一个 AI 算法模型的实例，建议从这里开始，详见介绍参考：
   https://doc.comake.online/IPU_Sigdoc_vS03.0.8_zh/module/OpenDLA/OpenDLA.html
2. 该云平台主要提供红框相关的 Docker 环境的功能：支持 onnx/caffe/tflite/tensorflow graphdef 等类型的模型框架用该平台做模型转换、量化、推理。IPU Tool Chain 的深入功能介绍，参考：https://doc.comake.online/IPU_Sigdoc_vS03.0.8_zh/index.html

2 使用介绍

2.1 账号注册

Comake 社区地址：https://www.comake.online/index.php

Figure 2 Comake 社区首页

注册的用户获取权限之后可以直接登录使用。未注册的开发者可以参考该篇文章完成注册：https://dev.comake.online/home/article/1324

2.2 功能介绍

点击图 2 的红框 输入注册的账号密码登录平台，如下图所示，主要功能是文件管理与 webSSH 功能，下面进行使用的介绍。

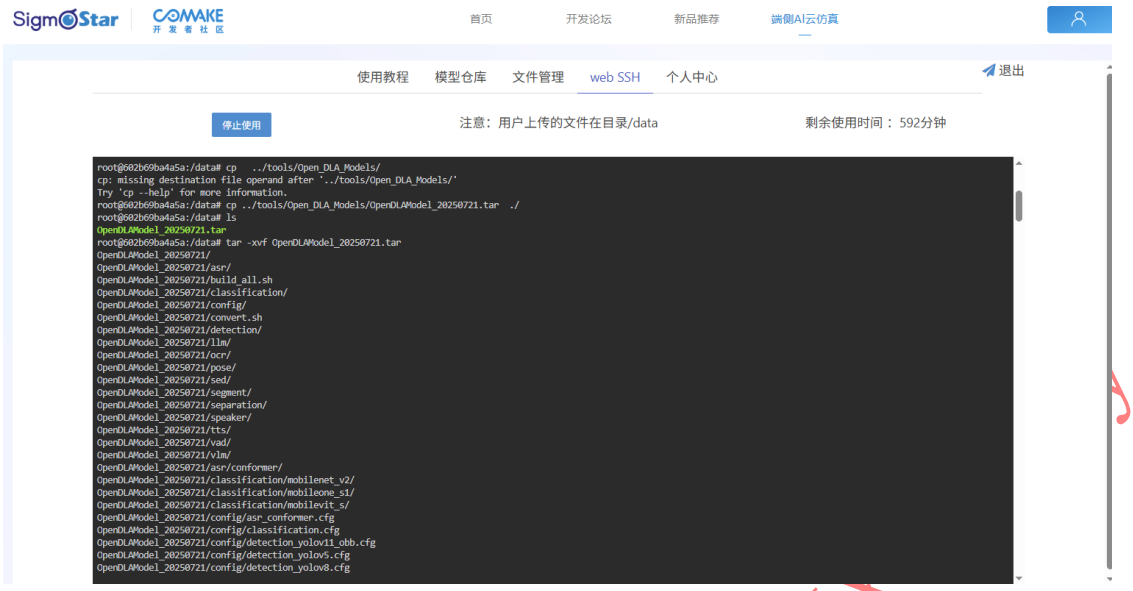


Figure 3 Comake 社区——AI 在线开发平台

2.2.1 文件管理

2.2.1.1 上传

在文件管理中，支持用户上传自己的模型文件，操作为：

1. 点击选择文件上传
2. 点击上传文件
3. 在/data/中显示上传文件

这些步骤对应下图中的 1，2，3

Figure 4 文件管理——上传文件

2.2.1.2 下载

在文件管理中，同样支持用户下载转换好的模型，操作为：

1. 点击/data/路径下对应的模型文件夹
2. 点击模型文件夹下对应的模型文件的下载按钮

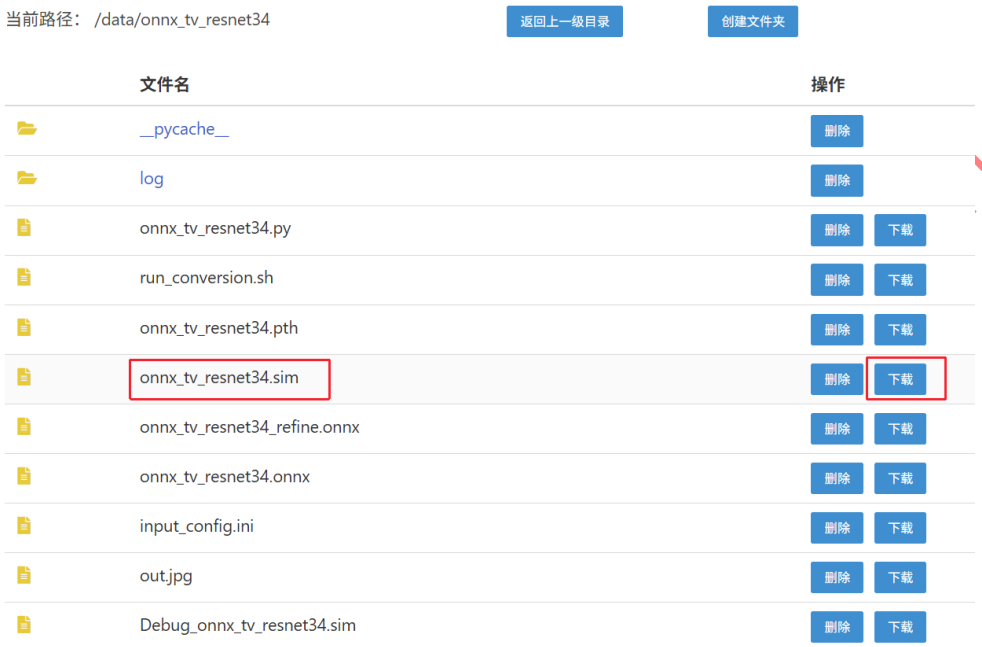


Figure 5 文件管理——下载文件

2.2.2 web SSH

web SSH 是云服务的远程登录窗口，点击 web SSH，再点击开始排队，等待 Server 空闲进入 SSH 界面，界面如下所示，已配合好 IPU Tool Chain 的使用环境，可以直接使用。

Figure 6 web SSH

特别注意：在离开不用的情况下， 请点击停止使用，否则会一直计时。

server 下重要的路径介绍：

- 用户上传文件路径为： /data/
- SGS_Models 资源存放路径为：/tools/SGS_Models/
- IPU_ToolChain 工具路径为：/tools/IPU_ToolChain/SGS_IPU_Toolchain_25070213/
- QuickStatrt 路径：/tools/IPU_ToolChain/Quick_Start_Demo/
- OpenDLAModel 路径：/tools/Open_DLA_Models/

2.2.3 模型仓库

在模型仓库中，提供了端侧 AI 云仿真所上传的 SGS Models、Open DLA Models、DLA Models 下载路径以及我司提供的模型列表，同时也提供了不同 AI 架构的模型列表，详情请看模型仓库页面。

Figure 7 模型仓库

2.3 使用示例

2.3.1 IPU_ToolChain 使用示例

本章节以 ONNX 框架 yolov8s 作为示例

2.3.1.1 配置环境

1.拷贝 yolov8s

```
#拷贝 yolov8s
cp tools/IPU_ToolChain/Quick_Start_Demo/onnx_yolov8s /data/
```

2.激活 SGS_IPU 的环境

```
cd tools/IPU_ToolChain/SGS_IPU_Toolchain_25070213/
source cfg_env.sh
```

3.查看支持芯片型号

```
python3 /tools/SGS_IPU_SDK_25070213/DumpDebug/show_sdk_info.py
```

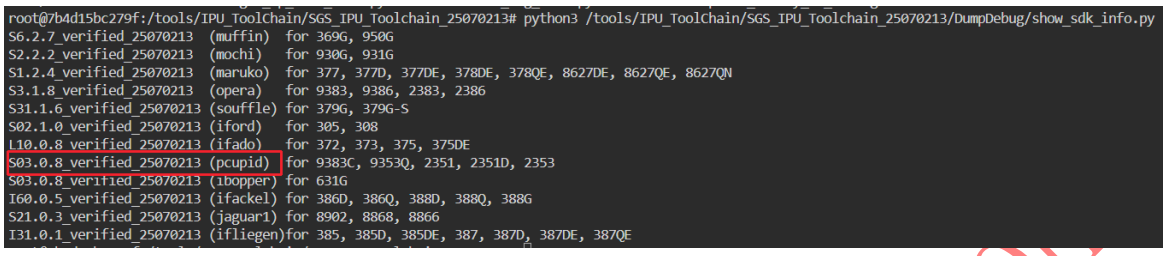


Figure 8 查看适配芯片

2.3.1.2 模型转换

执行以下命令进行模型的转换

```
cd /data/onnx_yolov8s/

python3 /tools/IPU_ToolChain/SGS_IPU_Toolchain_25070213/Scripts/
ConvertTool/ConvertTool.py onnx \
--model_file ./onnx_yolov8s.onnx \
--input_shape 1,3,640,640 \
--input_config ./input_config.ini \
--output_file ./onnx_yolov8s_float.sim \
--soc_version pcupid
```

转换结果如下：

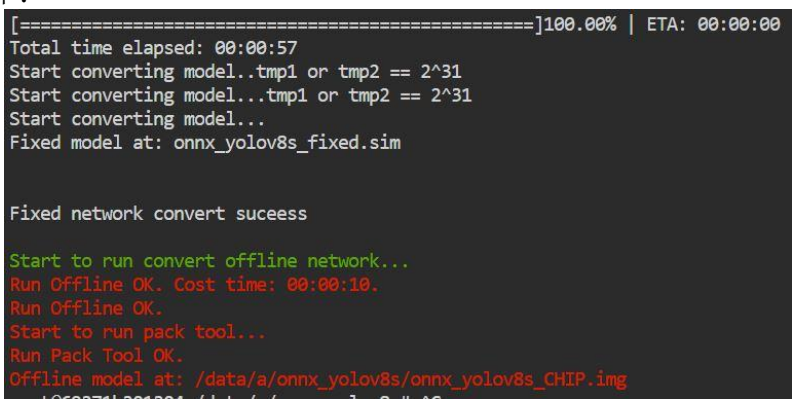


Figure 9 模型转换

2.3.1.3 仿真运行

PC 端仿真运行 float 模型

输入以下命令进行 PC 端仿真运行 float 模型

```
python3 yolov8_simulator.py \
--image 000000562557.jpg \
```

```
--model onnx_yolov8s_float.sim \
-n onnx_yolov8s_preprocess.py \
--draw_result output \
--soc_version pcupid
```

运行完成

```
Input(0):
    name:       images
    index:      0
    dtype:      <class 'numpy.float32'>
    layouts:    NHWC
    size:       4915200
    shape:      [1, 640, 640, 3]
Output(0):
    name:       output0
    index:      0
    dtype:      <class 'numpy.float32'>
    size:       2822400
    shape:      [1, 84, 8400]
)

[==================================================]100.00% | ETA: 00:00:00
Total time elapsed: 00:00:00
```

Figure 11 PC 端仿真运行 float 模型

结果如下：

Figure 12 PC 端仿真运行 float 模型结果

更多详细内容请参考 IPU_ToolChain 使用文档

https://doc.comake.online/IPU_Sigdoc_vS03.0.8_zh/index.html

2.3.2 OpenDLAModel 使用示例

OpenDLAModel&OpenDLA 是为用户提供在 sgs 平台上进行快速部署的开发套件。**更多详细内容请参考 `OpenDLAModel` 使用文档**

https://doc.comake.online/IPU_Sigdoc_vS03.0.8_zh/module/OpenDLA/OpenDLA.html

本小节以 OpenDLAModel 下的 yolov8 为例，简单介绍用法。

2.3.2.1 配置环境

```
#拷贝 OpenDLAModel
cp -r  tools/Open_DLA_Models/OpenDLAModel_20250721 /data/
```

2.激活 SGS_IPU 的环境

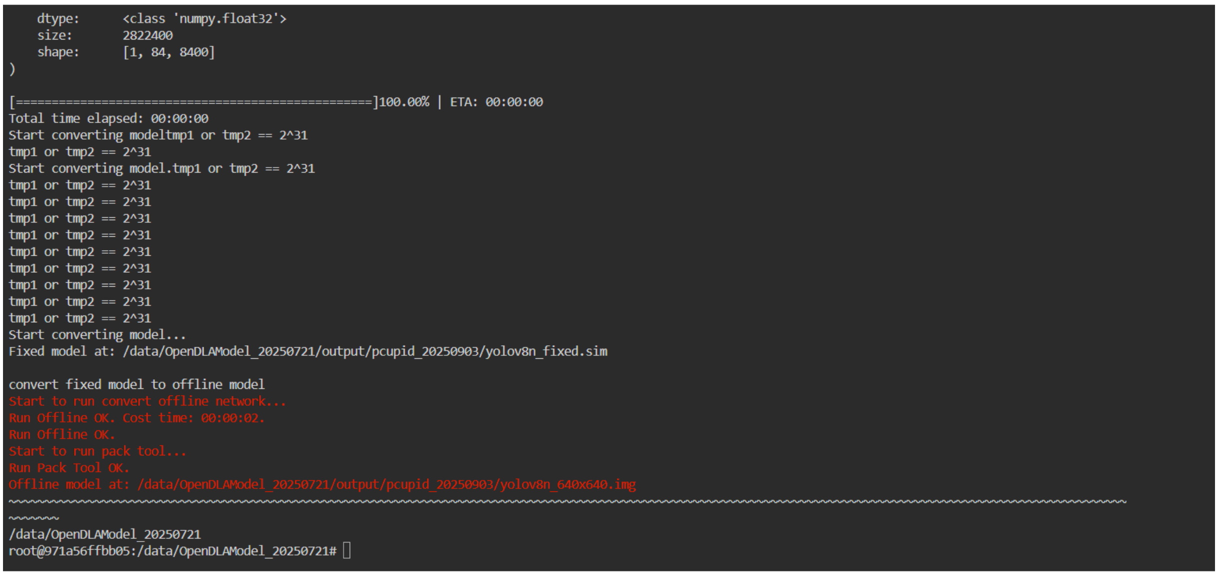
```
cd tools/IPU_ToolChain/SGS_IPU_Toolchain_25070213/
source cfg_env.sh
```

2.3.2.2 模型转换

执行以下命令进行模型的转换

```
cd OpenDLAModel_20250721
bash convert.sh -a detection/yolov8 -c config/detection_yolov8.cfg -p
/tools/IPU_ToolChain/SGS_IPU_Toolchain_25070213/ -s false
```

转换结果如下：



```
    dtype:      <class 'numpy.float32'>
    size:       2822400
    shape:      [1, 84, 8400]
)

[==================================================]100.00% | ETA: 00:00:00
Total time elapsed: 00:00:00
Start converting modeltmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
Start converting model.tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
tmp1 or tmp2 == 2^31
Start converting model...
Fixed model at: /data/OpenDLAModel_20250721/output/pcupid_20250903/yolov8n_fixed.sim

convert fixed model to offline model
Start to run convert offline network...
Run Offline OK. Cost time: 00:00:02.
Run Offline OK.
Start to run pack tool...
Run Pack Tool OK.
Offline model at: /data/OpenDLAModel_20250721/output/pcupid_20250903/yolov8n_640x640.img
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~
/data/OpenDLAModel_20250721
root@971a56ffbb05:/data/OpenDLAModel_20250721# 
```

Figure 13 模型转换

2.3.2.3 仿真运行

PC 端仿真运行 float 模型

输入以下命令进行 PC 端仿真运行 float 模型

```
bash convert.sh -a detection/yolov8 -c config/detection_yolov8.cfg -p
/tools/IPU_ToolChain/SGS_IPU_Toolchain_25070213/ -s true
```

运行完成，仿真结果会默认将 `float` 模型的输出 tensor 保存到存放在 `detection/yolov8/log/output/ *.txt` 文件里

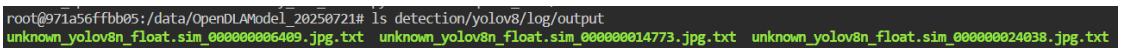


Figure 14 PC 端仿真运行 float 模型

通过 2.2.2 描述的文件下载，下载仿真结果文件，如下图所示

Figure 15 PC 端仿真运行结果

output0 Tensor:
{
tensor dim:3, Original shape:[1 84 8400]
tensor data:
3.239515 16.408733 22.294079 28.323288 33.302505 39.717018 47.357391 53.250008 62.516808 70.895958 91.9734
129.227844 135.357605 148.790344 157.874634 159.664917 177.826324 181.171143 190.503372 202.041412 203.141
257.708923 262.474152 265.503784 280.612640 294.681824 300.386719 311.667358 314.531067 326.501465 334.476
395.639832 399.427002 403.464874 412.175537 410.459900 413.538940 429.827271 451.693909 460.709534 465.449
519.018860 521.546936 531.584656 536.683289 540.887817 552.875244 563.705994 576.897949 584.707031 589.479
3.126708 15.665174 23.892830 31.372442 34.711754 39.549881 43.489006 47.991539 58.712318 68.034157 92.5089
128.979828 134.682617 151.505249 158.779541 160.911942 181.877258 184.173813 185.757507 200.733322 202.189
259.188019 260.050781 262.820496 284.428101 287.623535 300.270081 311.908997 313.451080 326.586639 330.376
394.153809 396.065857 402.852356 411.540283 412.091492 428.917633 431.956665 444.862915 459.603760 461.644
517.984863 518.978699 529.935242 536.485107 549.873657 552.188477 564.587769 575.709961 586.285278 592.822
2.818982 14.943662 24.951279 32.830597 35.092068 39.433868 43.174660 48.453979 58.930321 68.188980 91.6976
131.464233 134.097198 152.170715 155.871674 171.982605 183.553040 184.234283 184.803314 199.629532 200.261
259.013000 259.965088 264.274750 286.624603 289.168640 301.631439 311.332886 312.923279 327.199677 328.097
393.652771 393.018616 400.133820 405.570251 406.263641 411.349457 434.711426 449.489471 458.605347 461.480
518.068176 519.245117 529.083069 538.215088 547.529907 551.138550 565.614746 575.006592 586.192505 593.644
2.827814 12.374537 23.022297 31.070414 33.965096 38.457962 44.595284 51.819134 60.894417 67.284973 92.7290
127.731102 138.151093 151.402954 155.458557 166.553482 183.853546 184.440765 185.080475 186.571182 200.665
257.201202 254.904907 258.749756 280.152344 290.661560 304.105865 311.081116 314.590881 326.489380 326.916
394.690308 399.640259 405.772156 405.290588 404.556824 425.146912 432.556732 450.274872 457.448792 460.979
517.129211 520.678223 527.991150 540.138000 546.455566 549.585327 566.277466 574.386597 583.704346 591.233

Figure 16 PC 端仿真运行结果