

EEP-TPU 库文件示例程序说明手册

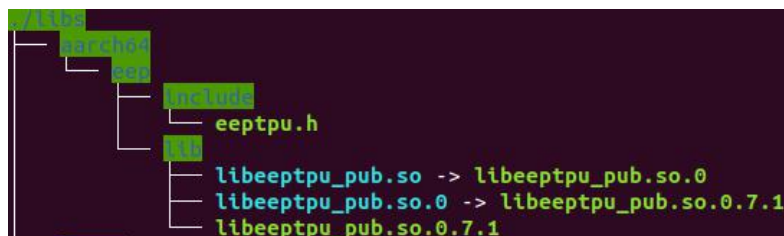
示例程序使用的库文件为：libeeptpu_pub.so; 头文件为：eeptpu.h。库文件存放在 libs 文件夹下，有 3 个不同平台的库，分别是 ARM32、AARCH64、X86。用户可根据实际情况选择其中的平台进行测试。在使用 EEP-TPU 编译器编译神经网络算法时，请使用参数“**--public_bin**”，以便生成后缀为“**.pub.bin**”的 bin 文件，本库文件需使用这种“**.pub.bin**”类型的 bin 文件。

Demo/libs 目录默认为空，请先把最新版本的库文件解压缩并拷贝到 demo/libs 目录

下。例如进入到 libeeptpu_pub 目录，打开命令行运行解压缩命令：

```
tar xzvf libeeptpu_pub_v0.7.1.tar.gz
cp -r libeeptpu_pub_v0.7.1/* ../demo/libs/
```

以 aarch64 平台为例，demo/libs 目录下的结构如下图所示，确保 eep/lib 目录下包含 3 个库文件，其中两个为链接文件。



每个示例程序有其自己的文件夹，例如：classify、icnet、multi_bins_test、nntpu_test、yolo 等，每个示例文件夹下都有对应的编译脚本和运行脚本。此外，common 文件夹包含了共同使用的代码（读图和读 npy 文件）；eeptpu_bins 包含预编译好的算法 bin 文件，请自行到 github 上本项目所在位置的 readme 里查找对应的下载地址，并将其拷贝到该文件夹下。

《EEP-TPU 应用开发接口（API）使用手册》里，有介绍该库文件的编译环境。请确

保您的使用平台下的编译环境与库文件编译环境相同，否则可能出现兼容性问题。

不同的硬件平台使用的 EEP-TPU 配置可能不一样，因此在代码中对 EEP-TPU 进行初始化时，要根据实际情况对配置进行修改，例如，寄存器配置 `eeptpu_set_tpu_reg_zones`、内存基址配置 `eeptpu_set_base_address` 等。

需注意的是，github 上面评估版本的 EEP-TPU 有**使用时间限制**，超过时间限制后，EEP-TPU 将不可用。表现在示例程序运行上，有可能是程序运行时卡死。

示例程序编译：

将所有示例程序的文件夹整体拷贝到目标平台下，根据平台选择如下编译脚本：

```
ARM32 平台：      ./compile.sh 32
```

```
AARCH64 平台：    ./compile.sh 64
```

```
X86 平台：         ./compile.sh 86
```

示例程序运行：

每个示例程序下有对应的运行脚本示例 `test.sh`。**在运行前，请确保 `eeptpu_bins` 文件夹下包含了对应的算法 bin 文件。**可在 github 本项目的介绍中，获取 `eeptpu_bins` 已编译 bin 文件的下载地址，下载完后将 bin 文件拷贝到该目录下。

以下是示例程序功能的简单介绍：

示例程序：**classify**

分类网络的测试程序。可测试分类网络，例如 `mobilenet`、`resnet`、`vgg`、`squeezenet`、

inception 等网络。

示例程序：**icnet**

ICNet 语义分割网络的测试程序。以 ICNet 为例，网络输入为 513x1025。网络推理结束后，可保存推理结果图片以供查看。

示例程序：**multi_bins_test**

单核多 bin 文件加载推理的测试程序。以 mobilenet 和 yolov4tiny 网络为例，单核加载这两个网络，并依次推理。

示例程序：**nntpu_test**

神经网络或算子通用测试程序。无特定网络的后处理代码，推理后可将推理结果输出到 test_output_data.txt 文本文件，以便跟该网络或算子的软件推理结果进行对比。支持 Pack 模式的输入数据。支持 npy 输入数据或者图像输入数据。

示例程序：**yolo**

目标检测网络的测试程序。可使用常用的目标检测神经网络来测试，例如：yolov3 系列/yolov4 系列/mobilenetyolov3/mobilenet-ssd 等。网络推理结束后，可保存推理结果图片以供查看。

其他示例程序

由于我们在 github 上面开放的 EEP-TPU 不是完全功能的，因此有些功能在此版本的 TPU 上无法使用。例如多核 TPU 的测试、参数复用、“Pack”精度转换模式的测试、算法跳转测试等，这些功能在 EEP-TPU 编译器使用手册上有更为详细的介绍。若需要体验这些商

业版功能，可与我司联系。

上述未提供代码的部分测试程序简介：

多核 TPU 测试： 单个网络可以编译并部署在多核 TPU 上进行推理，推理时间会比单核 TPU 快。

参数复用： 多核 TPU 下，若同一网络需要被多个 TPU 核使用时，可采用参数复用形式加载算法，使算法参数仅加载一次就可以被多个 TPU 使用。减少内存占用空间。

Pack 精度转换模式： 输入数据支持 FP32/FP16/INT8/INT16/INT32/UINT8 格式。

例如 opencv 读图时，cv::Mat 所存储的 data 数据可以不用转换而直接给 TPU 使用。

算法跳转： 在 TPU 层面上，从算法 1 直接跳转到算法 2，算法 1 的推理结果作为算法 2 的输入。