**算法模型在嵌入式平台上的部署**

———**以RK3588s为例**

ROC-RK3588S-PC，采用Rockchip RK3588S新一代八核64位处理器，最大可配32GB大内存；支持8K视频编解码；支持NVMe SSD硬盘扩展；支持多种操作系统；可适用于边缘计算、人工智能、云计算、虚拟/增强现实等领域。

RK3588S是Rockchip全新一代旗舰AIoT芯片，采用了8nm LP制程；搭载八核64位CPU，主频高达2.4GHz；集成ARM Mali-G610 MP4四核GPU，内置AI加速器NPU，可提供6Tops算力，支持主流的深度学习框架。参见[ROC-RK3588S-PC](https://www.t-firefly.com/product/industry/rocrk3588spc)。

NPU（Neural-network Processing Units）可以说是为了嵌入式神经网络和边缘计算量身定制的，但若想调用RK3588s的NPU单元进行推理加速，则需要首先将模型转换为.rknn格式的模型。

# 1、过程概述

这次我们将yolov5模型部署在RK3588s上，并使用NPU推理。过程分以下几步：

1.使用正确版本（v5.0）的yolov5进行训练得到pt模型。

2.将pt模型使用yolov5工程中的export.py转换为onnx模型。（pt->onnx）

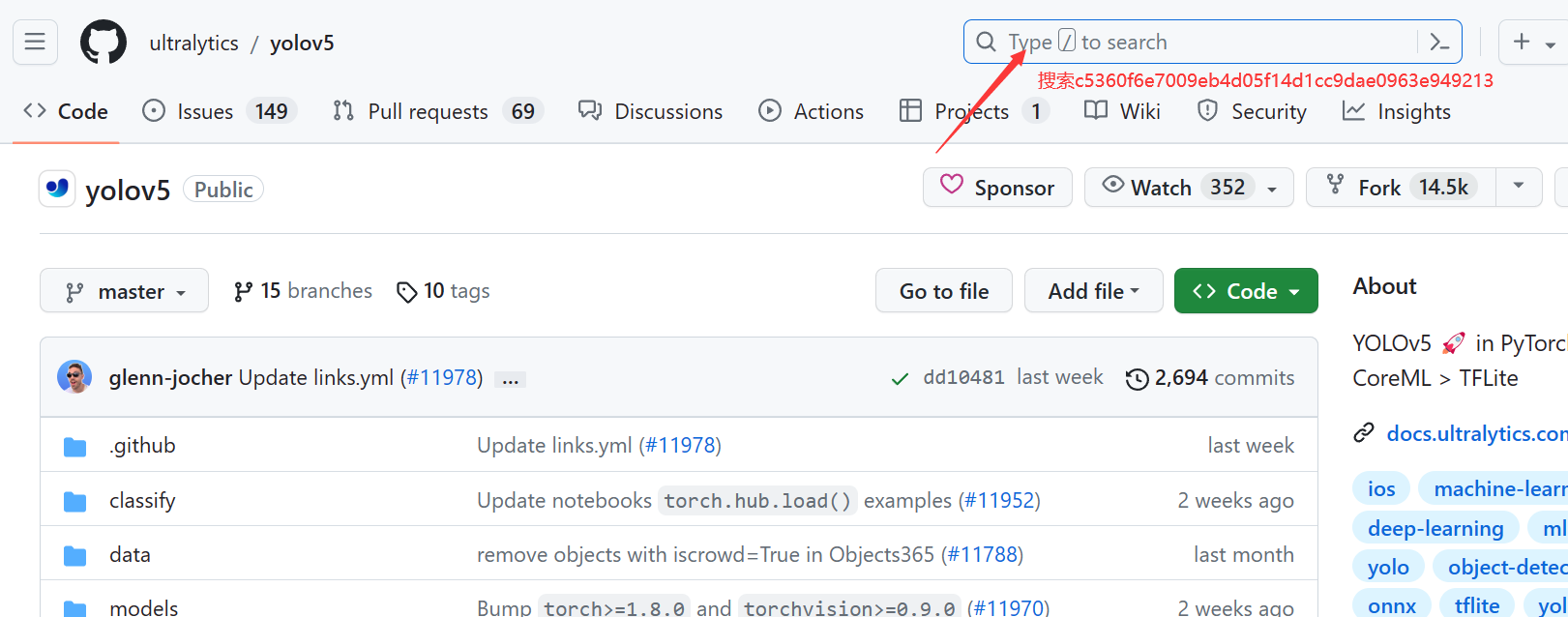
3.将onnx模型使用rknn-toolkit2中onnx文件夹的test.py转换为rknn模型。(onnx->rknn)

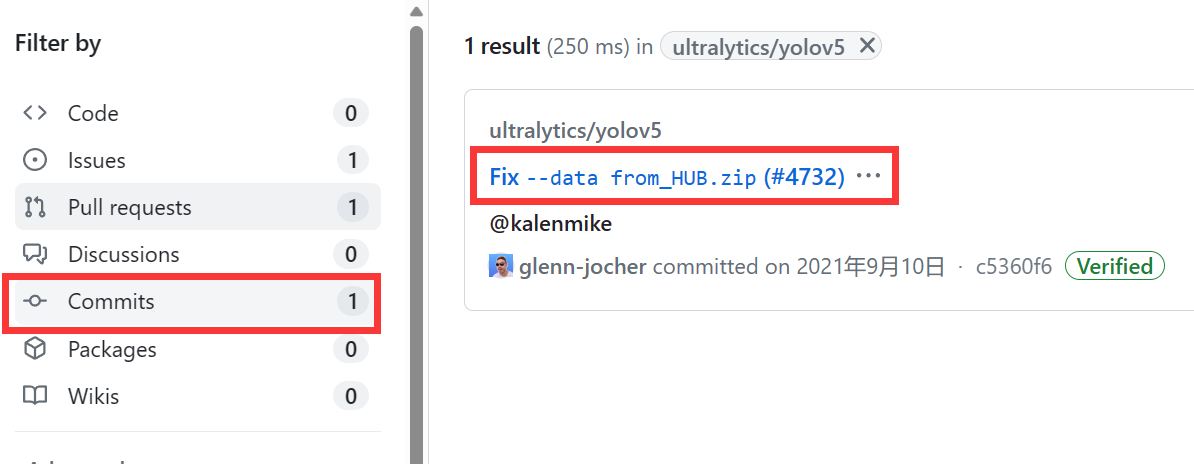
4.在板子上使用rknpu2工具调用rknn模型，实现NPU推理加速。

# 2、部署过程

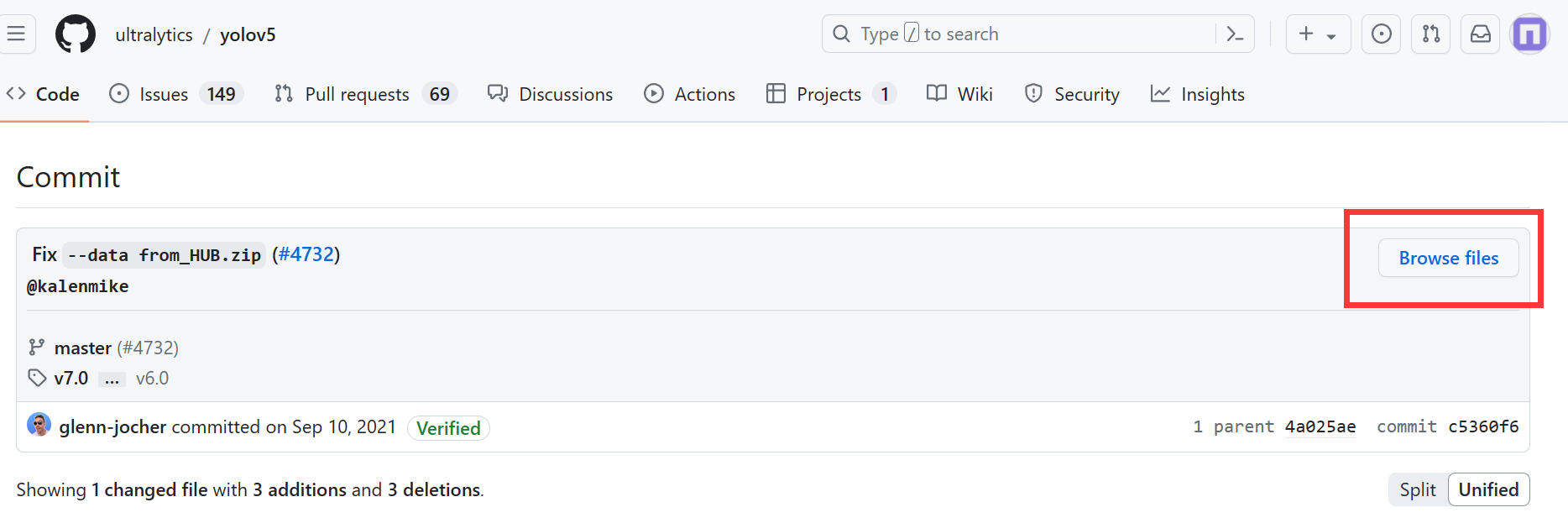
## 2.1、训练yolov5模型(平台：windows)

注意需要使用特定版本的yolov5。进入[yolov5官网](https://github.com/ultralytics/yolov5)，搜索c5360f6e7009eb4d05f14d1cc9dae0963e949213。

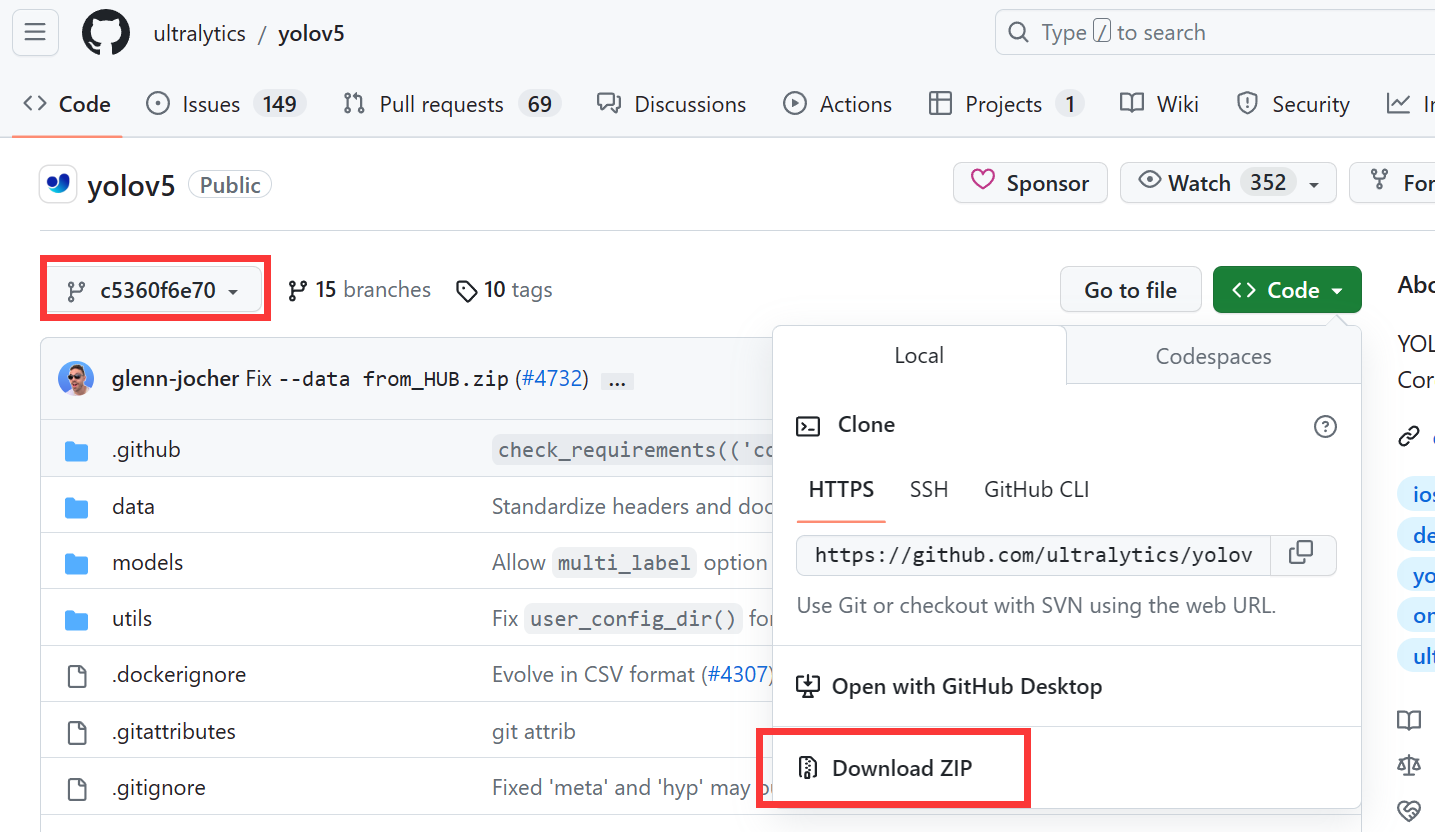
找到特定版本的commit。



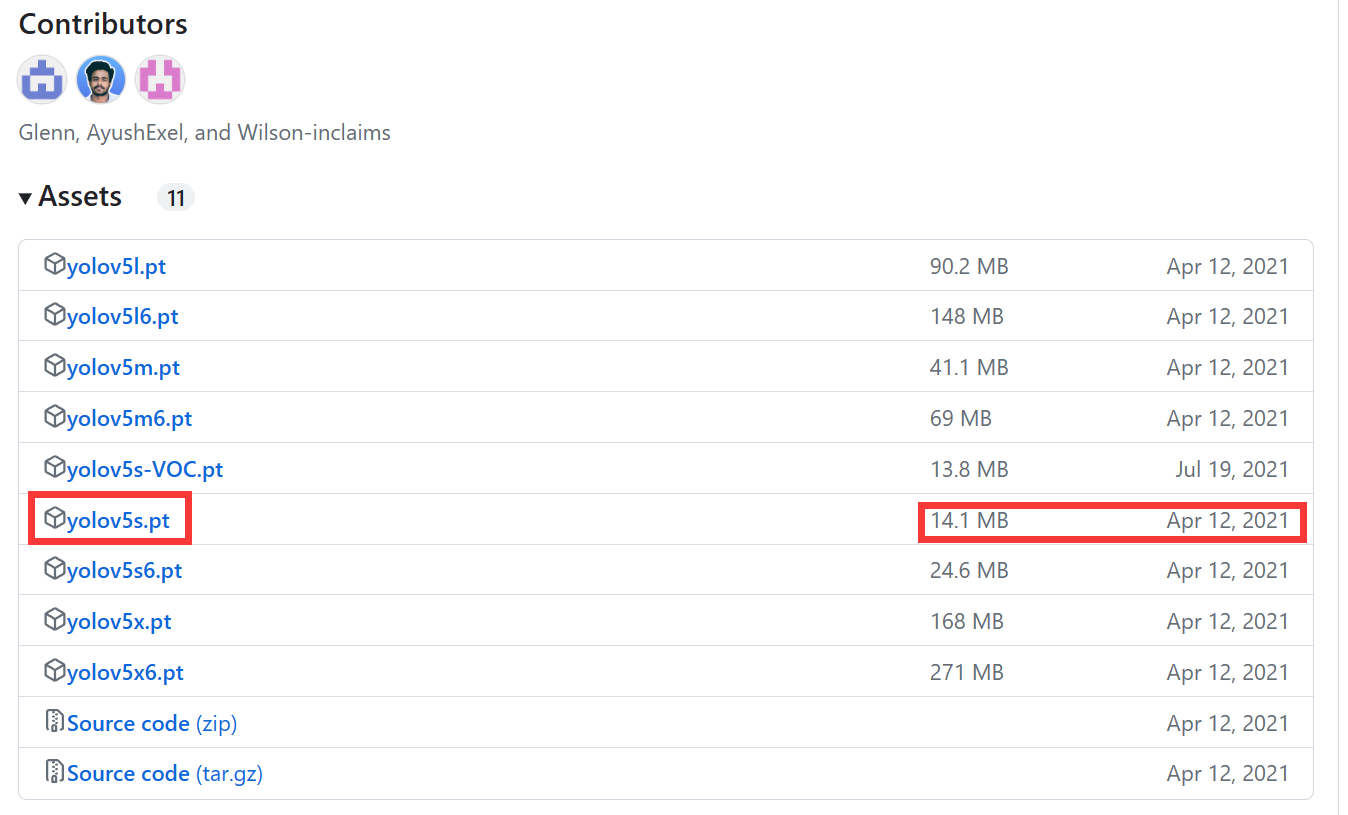
点击Browse files去到特定版本yolov5源码。



接下来，点击download zip，注意不要git clone，那样还是最新版的yolov5版本。



代码下载好后，开始训练自己的pt模型；此处实际上是yolov5 5.0版本的工程，其中未包含预训练模型yolov5s.pt，需要自行下载，在[releases链接](https://github.com/ultralytics/yolov5/releases)中找到V5.0下载yolov5s.pt：



接下来就是配置好yolov5所需的环境，用自己的数据集训练yolov5模型，得到目标模型best.pt。

## 2.2 best.pt转换为best.onnx（平台：windows）

修改一段代码（注意，训练阶段不要修改）。将yolov5/models/yolo.py文件中class Detect(nn.Module)类的函数forword修改。



修改后将best.pt移动至yolov5/export.py同一文件夹下，在命令行调用以下命令：

python export.py --weights best.pt --img 640 --batch 1 --include onnx

便可以得到成功转换的模型best.onnx。

(可能会报错说ONNX版本不支持，那么就将export.py中parse\_opt()函数的--opset参数修改为12)

(也可能报错说不支持SiLU，把./models/common.py里的SiLU改成ReLU即可)

## 2.3 best.onnx转换为best.rknn（平台：Linux）

此处的转换工具只能在Linux系统上运行，在linux上下载（clone）[转换工具rknn-toolkit2](https://github.com/rockchip-linux/rknn-toolkit2)。并配置好相关环境，使用docker或直接配置。

1. Docker

根据rknn-toolkit2\docker\docker\_file\ubuntu\_18\_04\_cp36目录下的Dockerfile构建。

1. 直接配置（我所采用的）

2.1 下载conda环境（有的话跳过）

wget -c <https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh>

chmod 777 Miniconda3-latest-Linux-x86\_64.sh

sh Miniconda3-latest-Linux-x86\_64.sh

结束后重启命令行或执行source ~/.bashrc。

* 1. 创建名为rknntool的python3.8环境，并进入：

conda create -n rknntool python=3.8

conda activate rknntool

2.3 进入./doc目录

pip install -r requirements\_cp38-1.5.0.txt

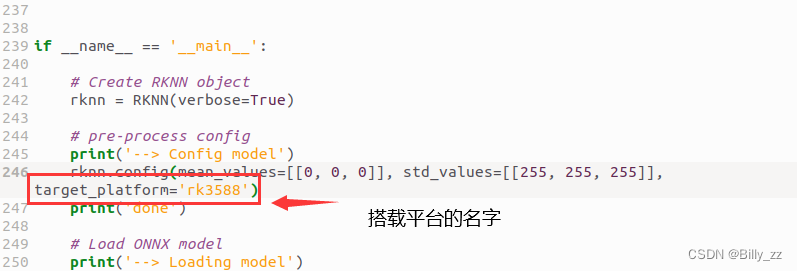
* 1. 进入./ packages目录

pip install rknn\_toolkit2-1.5.0+1fa95b5c-cp38-cp38-linux\_x86\_64.whl

终端中运行python，输入from rknn.api import RKNN，不报错即成功。

进入./examples/onnx/yolov5，将上文得到的best.onnx文件和待预测图片（也可以没有）复制到此文件夹中，修改test.py文件。（还）





修改完后执行python test.py，这里的test.py构建了一个虚拟的NPU运行环境，模拟在RK3588上真实运行的情况。

这时在当前文件夹./example/onnx/yolov5中可以看到生成的best.rknn模型和结果图片result.jpg。

## 2.4 RK3588s部署实现NPU加速（平台：开发板Linux系统）

开发板原生是安卓系统，如果用户要运行其他操作系统，需要使用对应的固件烧写到主板。完整流程及各种开发指南参见：[RPC-RKL3588S-PC](https://wiki.t-firefly.com/zh_CN/ROC-RK3588S-PC/)

### 2.4.1 换系统

#### 2.4.1.1 准备

需要：

* ROC-RK3588S-PC 开发板
* [固件](https://www.t-firefly.com/doc/download/164.html)
* 主机（本机windows）
* 良好的Type-C 数据线

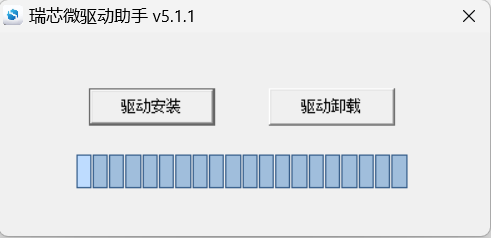
固件一栏中Ubuntu固件，里面版本有点多，有简装版和桌面版等。~~我所下载的是Ubuntu>Ubuntu20.04>Minimal中的ROC-RK3588S-PC\_Ubuntu20.04-Minimal-r2407\_v1.1.1a\_230704.7z压缩包~~（学习用途sb才下Minimal，直接桌面版）。下载好后解压，改名为update.img。

#### 2.4.1.2 安装烧写工具

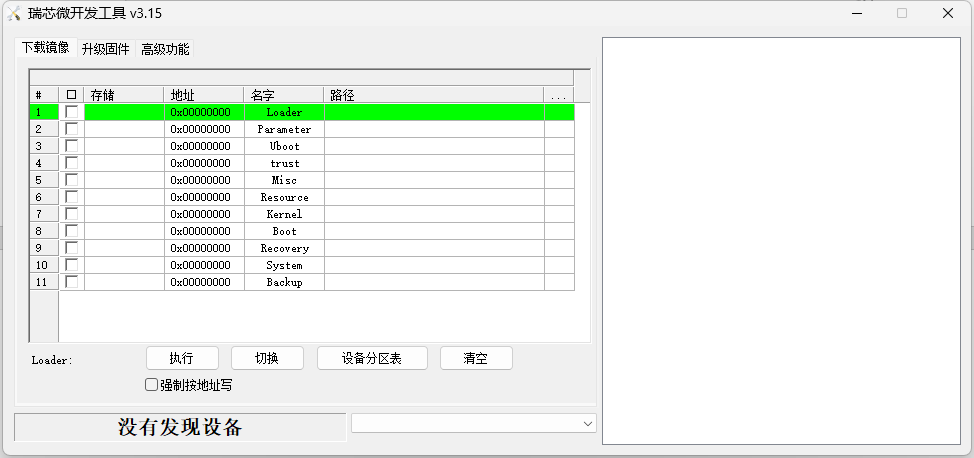
进入[Release\_DriverAssistant.zip](https://www.t-firefly.com/doc/download/164.html#other_572)，下载里面部分文件。



解压，然后运行DriverAssitant里面的DriverInstall.exe 。为了所有设备都使用更新的驱动，请先选择驱动卸载，然后再选择驱动安装。

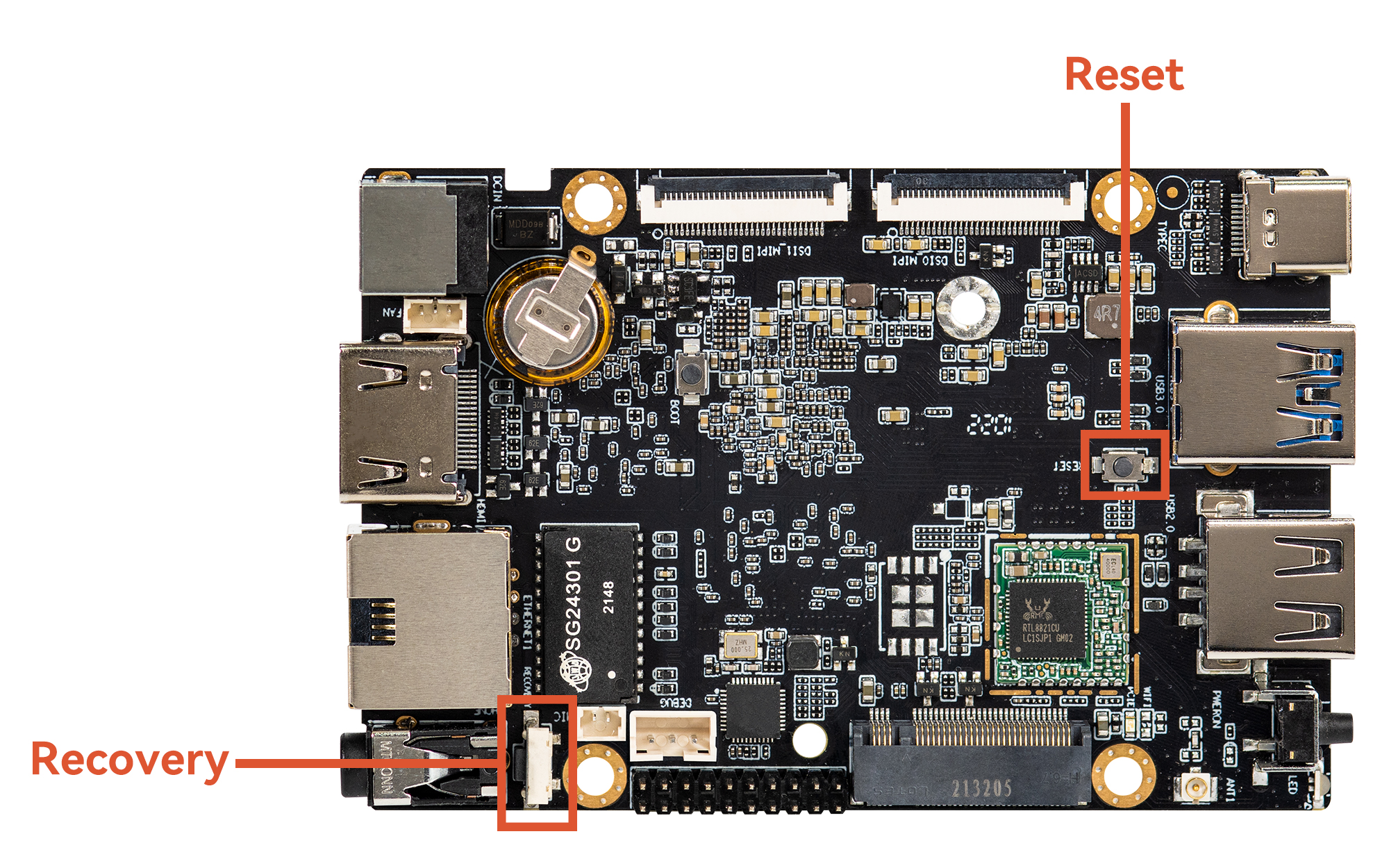


然后运行RKDevTool\_v3.15\_for\_window文件夹里的RKDevTool.exe，显示如下：



#### 2.4.1.3 进入升级模式

1. 先断开电源适配器连接
2. 使用 Type-C 数据线一端连接主机，一端连接开发板
3. 按住设备上的 RECOVERY （恢复）键并保持



1. 接上电源
2. 大约两秒钟后，松开 RECOVERY 键

通过RKDevTool.exe工具可以看到下方提示Found One LOADER Device。（上方是啥不管）

#### 2.4.1.4 烧写固件

1. 切换至Upgrade Firmware（升级固件）页。
2. 按Firmware（固件）按钮，打开要升级的固件文件（刚刚的update.img）。升级工具会显示详细的固件信息。
3. 按Upgrade（升级）按钮开始升级，等待右侧成功完成即可。

此时系统就换完了，重新上电后就是以ubuntu系统启动，可以连接hdmi进入可视化的命令行操作。

### 2.4.2 网络配置（平台：开发板系统）

（桌面版直接连WIFI或者热点，再用本机windows ssh连接）Ubuntu Minimal系统开机启动后，自动登录到 root 用户，密码为 firefly。其他系统参见：[ROC-RK3588S-PC](https://wiki.t-firefly.com/zh_CN/Firefly-Linux-Guide/first_use.html)。

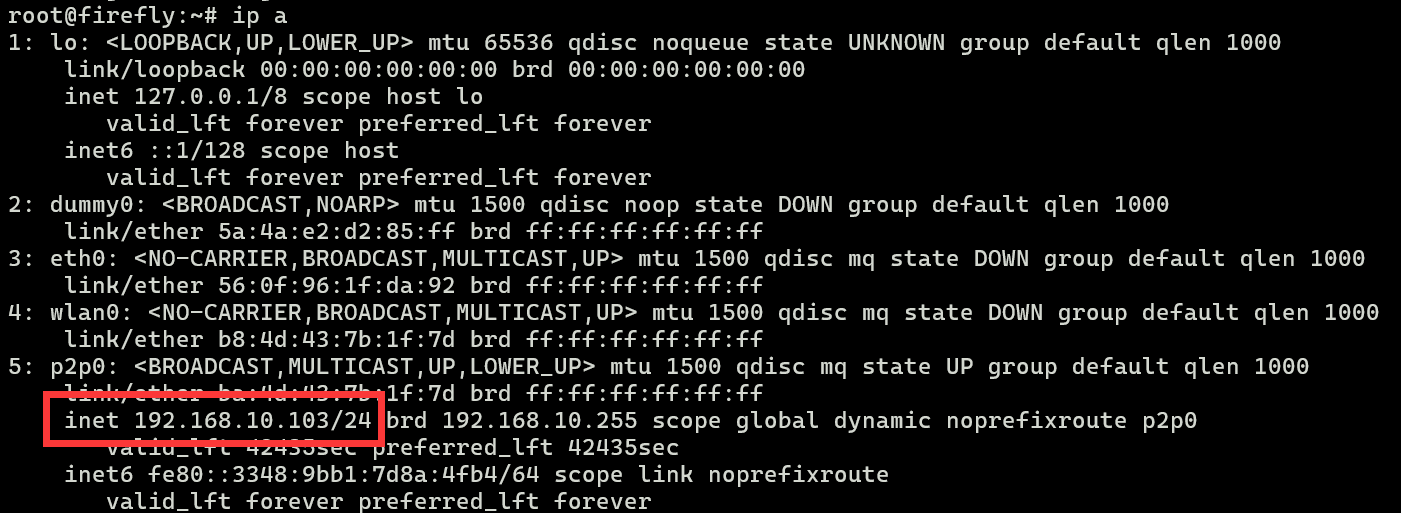
我所配置的是wifi网络，重启后会自动连接。更多种网络配置参见：[Firefly Linux 开发指南](https://wiki.t-firefly.com/zh_CN/Firefly-Linux-Guide/index.html)。

依次执行图中指令：



最后的指令是：nmcli dev wifi connect WIFI名 password WIFI密码

可用ip a指令查看当前的ip地址，以及ping指令验证是否网络配置成功。

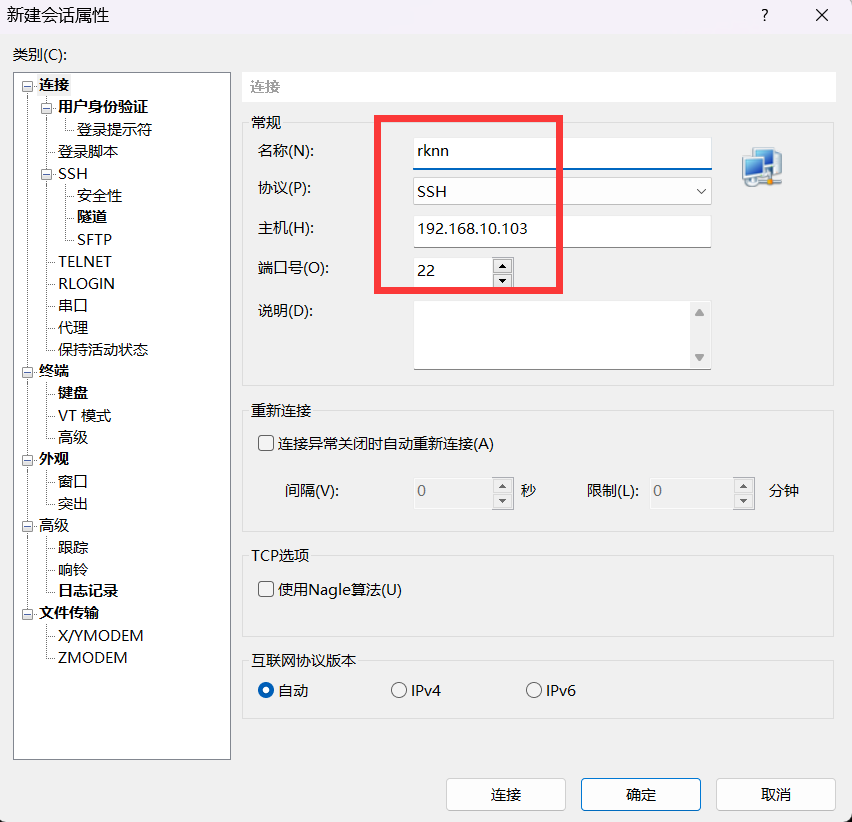


成功后需要开启ssh服务方便操作开发板，参见：[ubuntu ssh服务](https://blog.csdn.net/qq_16102655/article/details/85340432)

### 2.4.3 ssh连接（平台： windows）

注意要与开发板在一个局域网下，使用XShell等软件或ssh指令或VSCode等。

1. 使用XShell新建会话：



2. 使用ssh指令：ssh 要登录的用户@ip地址

例如：ssh [root@192.168.10.103](mailto:root@192.168.10.103)。然后输入密码即可。

完成后即可在本机操作开发板（可能会有卡顿）。

### 2.4.4 部署

现在开发板环境还不完备，后面需要什么就安装什么，apt install xxx，例如git，gcc，g++等。

1. 拉取官方demo：[rockchip-linux/rknpu2](https://github.com/rockchip-linux/rknpu2)到自己的目录下

git clone <https://github.com/rockchip-linux/rknpu2>

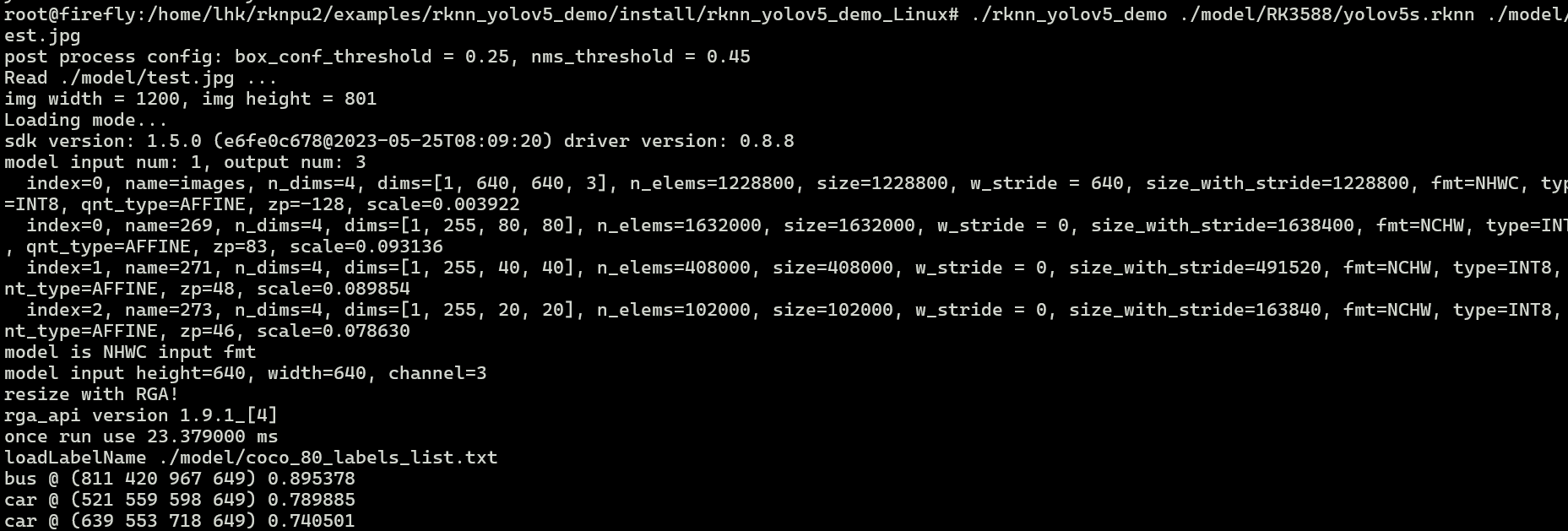
(注意此步骤不能windows上下载后传到开发板，要在开发板上拉取)

2. 进入/home/lhk/rknpu2/examples/rknn\_yolov5\_demo目录。有需要的话则修改下图内容为自己的。

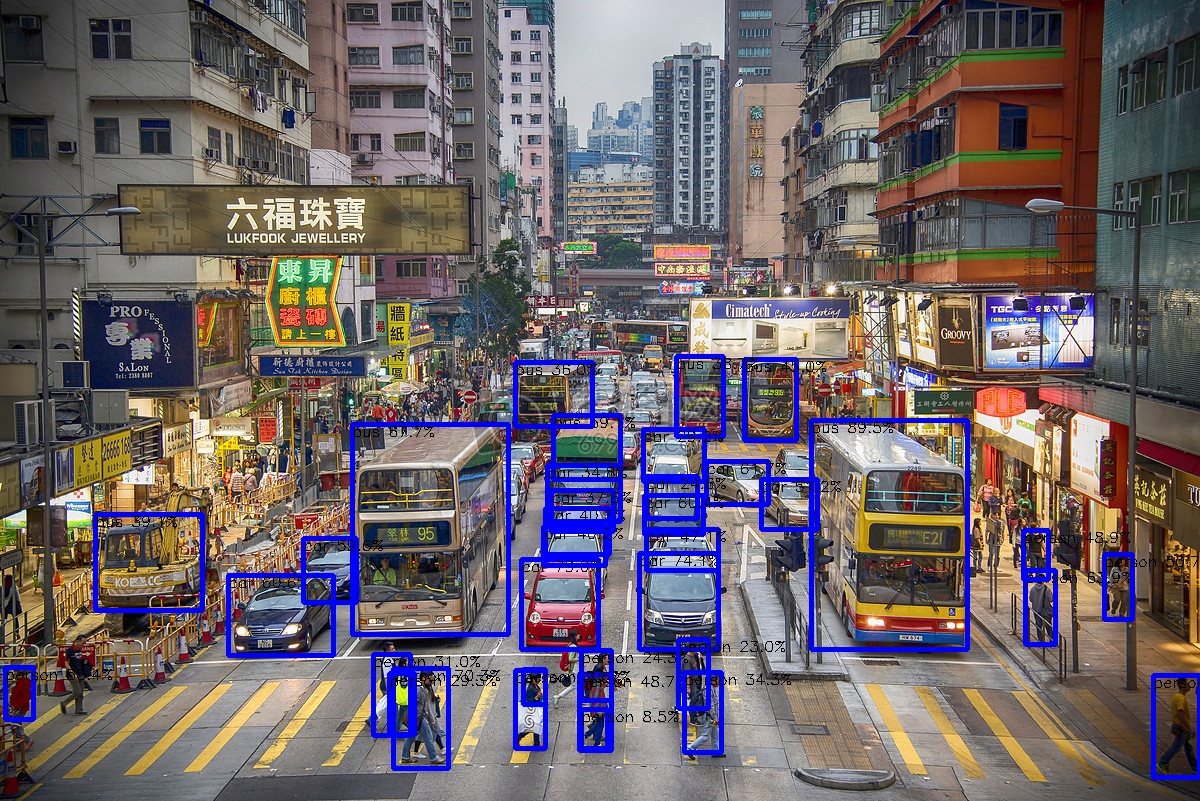


1. 将转换后的best.rknn文件放在model/RK3588目录下，运行bash ./build-linux\_RK3588.sh，成功后生成install文件夹，进入install/rknn\_yolov5\_demo\_Linux。（需要gcc，g++环境，安装即可）
2. 在model目录下放入需要推理的图片test.jpg，运行./rknn\_yolov5\_demo ./model/RK3588/best.rknn ./model/test.jpg

能看到命令行中：



在当前文件夹下生成out.jpg，例如下：



此时完成了单张图片在开发板上的推理。

也可以支持h264、h265、rtsp视频流。

1. 例如h264视频流，指令:

./rknn\_yolov5\_video\_demo model/RK3588/best.rknn model/test.h264 264

在当前文件夹下生成out.h264文件。([🡪out.mp4](file:///D:\桌面\rk\out.mp4))

注意需要使用h264码流视频，可以使用如下命令转换得到：

ffmpeg -i xxx.mp4 -vcodec h264 out.h264

2.例如rtsp视频流，指令：

./rknn\_yolov5\_video\_demo model/RK3588/best.rknn <RTSP\_URL> 265

### 2.4.5 性能

while true ; do cat /sys/kernel/debug/rknpu/load ; sleep 1 ; done;

~~pidstat -r -p <PID> 1 > memory\_data.txt~~