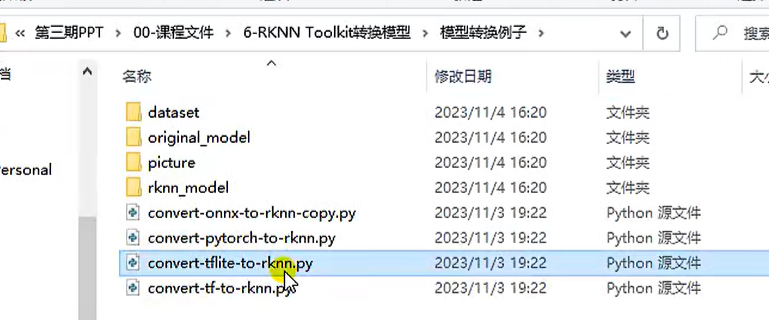


我们先来讲解将TFLite的模型转化为RKNN模型。参考视频资料文件：convert-tflite-to-rknn.py



见：

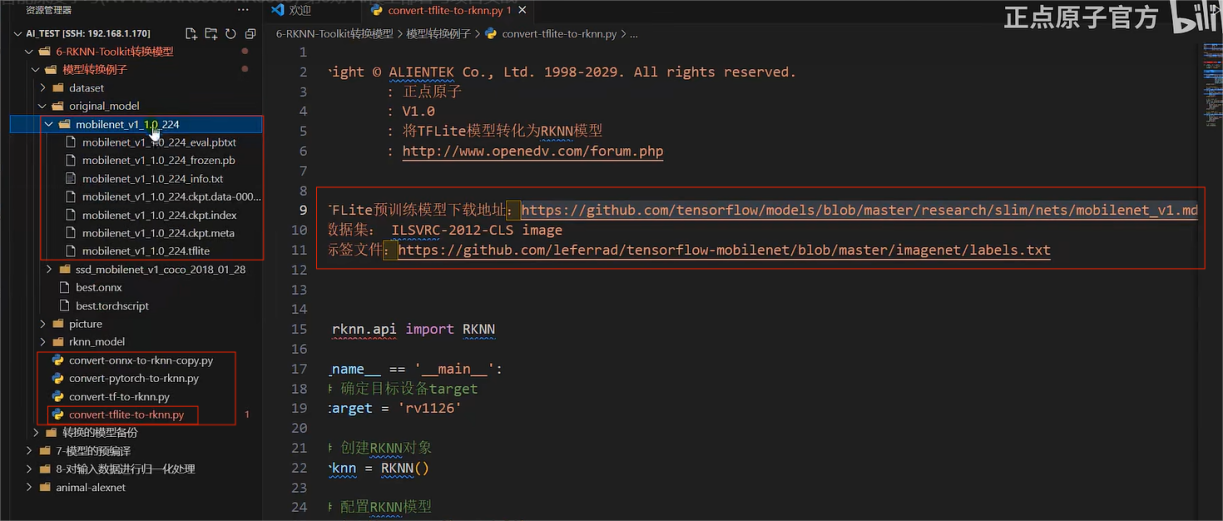
"C:\Users\zhongqing\Desktop\笔记\AI人工智能学习\_模型部署笔记\RK3568\_AI\正点原子RK3568 、RK3588 AI开发板\6-RKNN Toolkit转换模型\模型转换例子\模型转换例子\convert-tflite-to-rknn.py"

当然也可以直接看Rknn\_toolkit2的示例代码。

/\*

我们打开Ubuntu，在桌面的test文件中打开正点原子的示例文件6：

/home/alientek/桌面/test/6-RKNN Toolkit转换模型



在预训练模型下载地址下载后，可以得到左侧的mobilenet\_v1\_1.0\_224文件夹。其中的mobilenet\_v1\_1.0\_224.tflite就是tflite模型。从这个文件的名字也可以看到，这个模型的输入文件大小就是224\*224的彩色图片，使用的数据集是：ILSVRC-2012-CLS image这个数据集共有1000类，如果加上背景类的话，就有1001类，标签文件可以参考图中链接文件。

TFLite预训练模型下载地址：

<https://github.com/tensorflow/models/blob/master/research/slim/nets/mobilenet_v1.md>

数据集： ILSVRC-2012-CLS image

标签文件：

<https://github.com/leferrad/tensorflow-mobilenet/blob/master/imagenet/labels.txt>

这个就是mobilenet\_v1\_1模型的基本信息。下面我们直接看代码。

代码里面模型的转化，就是PPT里的流程。

/\*

代码分析见：

"C:\Users\zhongqing\Desktop\笔记\AI人工智能学习\_模型部署笔记\RK3568\_AI\正点原子RK3568 、RK3588 AI开发板\6-RKNN Toolkit转换模型\模型转换例子\模型转换例子\convert-tflite-to-rknn.py"

\*/

/\*

运行测试convert-tflite-to-rknn.py文件：

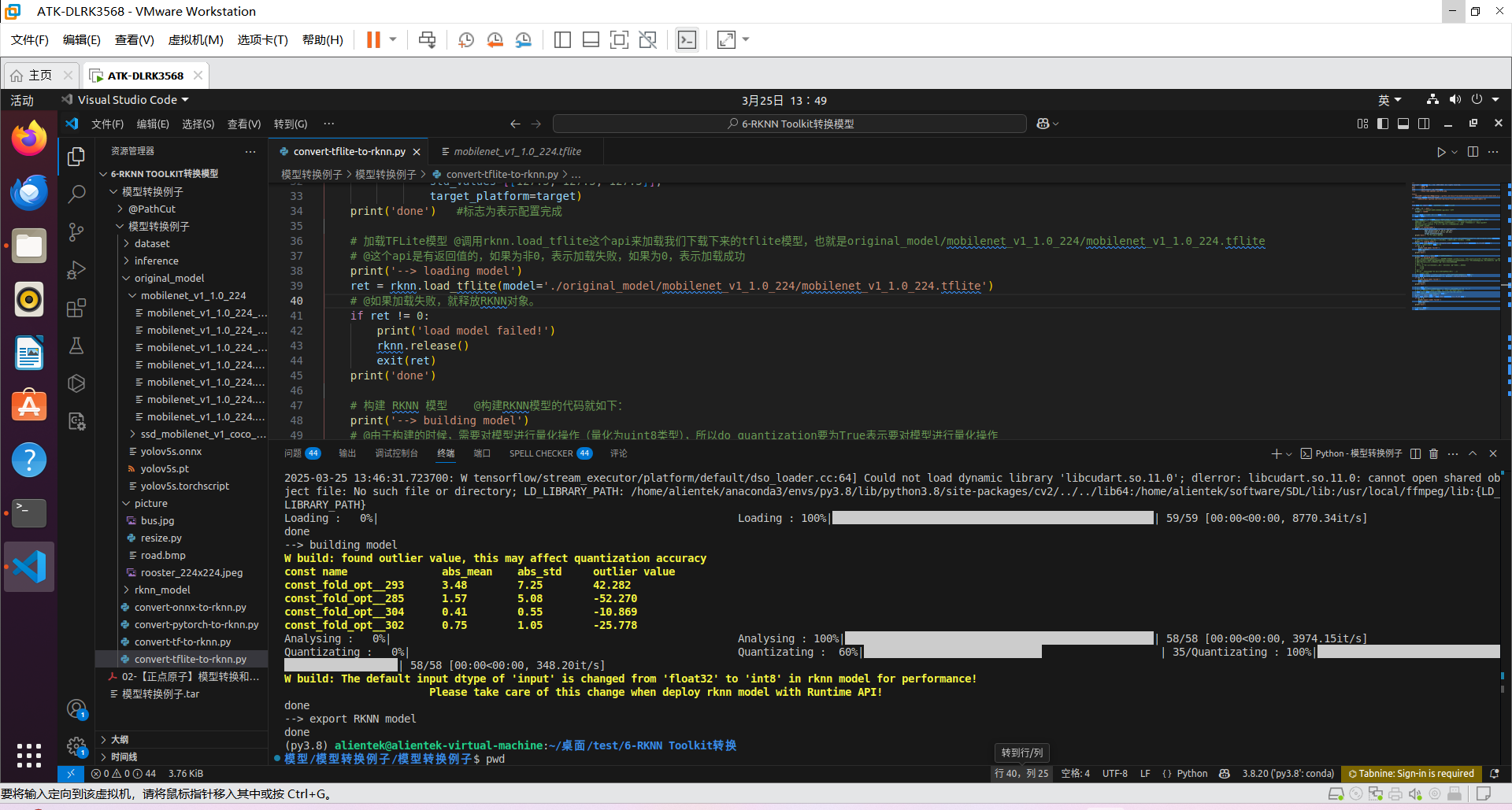
Ctrl+ shift + P选择环境,然后到对应的

/home/alientek/桌面/test/6-RKNN Toolkit转换模型/模型转换例子/模型转换例子

目录下运行：

python convert-tflite-to-rknn.py

运行测试结果如下：



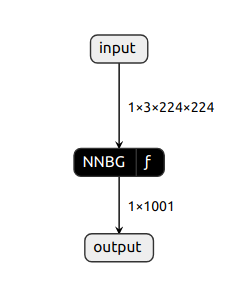
这里只是将Tflite文件转化为RKNN模型，只是进行模型转换，不需要运行模型，所以Ubuntu不需要通过ADB连接开发板。

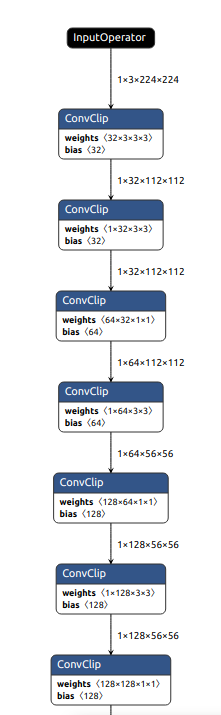
\*/

/\*

得到的RKNN模型位于rknn\_model目录下。得到的RKNN模型文件我们可以使用Netron进行对比分析：

IMG_256





得到这个模型之后，后续我们可以调用RKNN Toolkit里的api/调用RKNPU里的C++接口的api来完成模型的推理。

\*/

/\*

代码：

"""

Copyright © ALIENTEK Co., Ltd. 1998-2029. All rights reserved.

作者 : 正点原子

版本 : V1.0

描述 : 将TFLite模型转化为RKNN模型

论坛 : http://www.openedv.com/forum.php

说明：

TFLite预训练模型下载地址：https://github.com/tensorflow/models/blob/master/research/slim/nets/mobilenet\_v1.md

数据集： ILSVRC-2012-CLS image

标签文件：https://github.com/leferrad/tensorflow-mobilenet/blob/master/imagenet/labels.txt

"""

from rknn.api import RKNN #@首先就是导入RKNN这个模块

if \_\_name\_\_ == '\_\_main\_\_':

# 确定目标设备target，@这个变量就是指定的目标平台

target = 'rk3568'

# 创建RKNN对象，@接着就是创建RKNN对象。

rknn = RKNN()

# 配置RKNN模型 @配置RKNN模型，配置的目的就是希望后期的RKNN模型能够按照如下的方式来处理和构建

#@指定了量化类型为 动态定点量化uint8,输入的均值：三个通道（RGB通道）的均值都是127.5

#@输入的标准差也是127.5 这两个成员就是对输入的数据进行归一化或标准化。将其归一化到-1到1之间

#@它没有reorder\_channel = '0 1 2'也就是说使用默认的通道顺序。

#@制定平台为rk3568

print('--> config model')

rknn.config(quantized\_dtype='asymmetric\_quantized-8',

mean\_values=[[127.5, 127.5, 127.5]],

std\_values=[[127.5, 127.5, 127.5]],

target\_platform=target)

print('done') #标志为表示配置完成

# 加载TFLite模型 @调用rknn.load\_tflite这个api来加载我们下载下来的tflite模型，也就是original\_model/mobilenet\_v1\_1.0\_224/mobilenet\_v1\_1.0\_224.tflite

# @这个api是有返回值的，如果为非0，表示加载失败，如果为0，表示加载成功

print('--> loading model')

ret = rknn.load\_tflite(model='./original\_model/mobilenet\_v1\_1.0\_224/mobilenet\_v1\_1.0\_224.tflite')

# @如果加载失败，就释放RKNN对象。

if ret != 0:

print('load model failed!')

rknn.release()

exit(ret)

print('done')

# 构建 RKNN 模型 @构建RKNN模型的代码就如下：

print('--> building model')

# @由于构建的时候，需要对模型进行量化操作（量化为uint8类型），所以do\_quantization要为True表示要对模型进行量化操作

# @dataset这个参数就是指定量化的时候要用到的数据集，要量化的数据集通过这个文件来指定。而这个文件只有一张图片。

# @如果这个dataset2.txt文件有多张图片，又该怎么写呢？

# 方式一：

# 表示a和b图片是单个输入的，也就是一次只输入一张图片来进行量化。

# a.png

# b.png

# 方式二：

# 如果想 一次输入两张图片 的话，可以将这两个写作一行：

# a.png b.png

# 构建RKNN模型的api也是有返回值的，返回非0表示失败。失败就需要释放RKNN对象

ret = rknn.build(do\_quantization=True, dataset='./dataset/dataset2.txt')

if ret != 0:

print('build model failed!')

rknn.release()

exit(ret)

print('done')

# 导出 RKNN 模型

# 构建好RKNN模型之后，就可以将模型导出了，导出的目的就是后续的开发使用

# 后续我们可以将导出的模型部署到开发板上。实现真正的落地应用，

# rknn.export\_rknn这个api的参数就是要导出模型的路径和名字。

# 返回非0表示失败。失败就需要释放RKNN对象

print('--> export RKNN model')

ret = rknn.export\_rknn('./rknn\_model/mobilenet\_v1\_1.0\_224.rknn')

if ret != 0:

print('export model failed!')

rknn.release()

exit(ret)

print('done')

# 最后释放RKNN对象

rknn.release()

\*/

\*/

下节讲解将TensorFlow里的模型转化为RKNN模型。