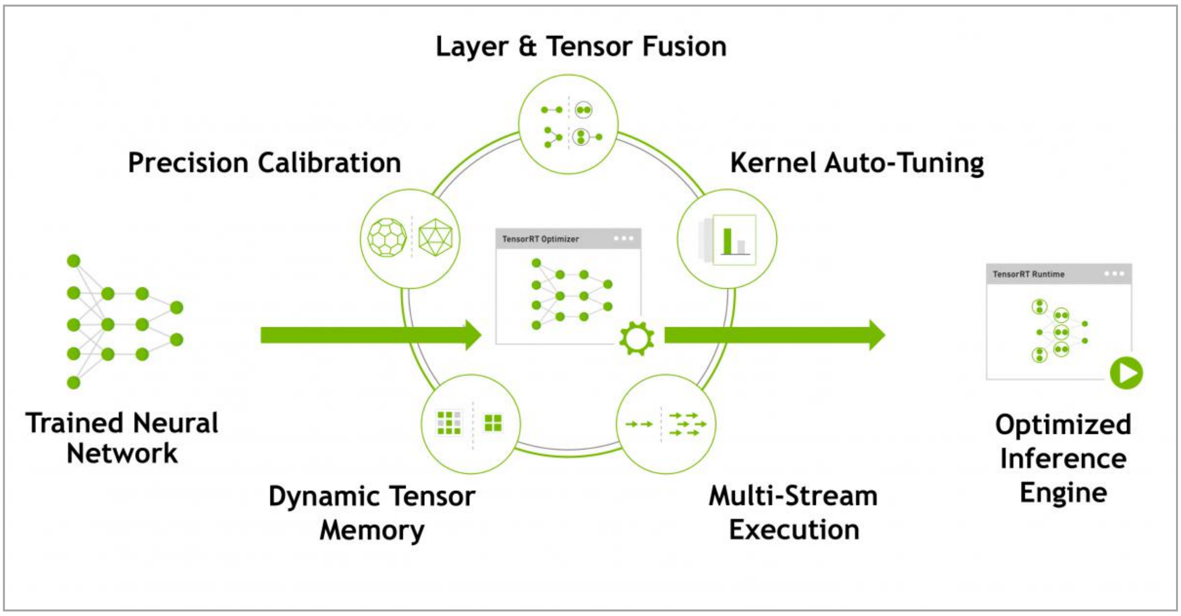
5.TensorRT Environmental construction（jetson-inference）



1. Preparation:

Because the model required for this example is about 1G, most of this example is not stored on the SD card (the SD card only has TensorRT required to run this model).

We need to remotely transfer the previously downloaded package to the corresponding directory.
First, if you don't have git and cmake, you need to install them.

```
sudo apt-get install libpython3-dev python3-numpy
sudo apt-get install git cmake
```

Clone the jetson-inference library

```
git clone https://github.com/dusty-nv/jetson-inference
```

Note:
If the system prompts, "error: RPC failed; curl 56 GnuTLS recv error (-54): Error in the pull function."
This reason is due to insufficient git default cache size.
We can increase the cache size using the following command

```
git config --global http.postBuffer 5242880000
```

If it still didn't work, it may be that the network speed is slow, configure the minimum speed and minimum speed time of git.

```
git config --global http.lowSpeedLimit 0
git config --global http.lowSpeedTime 999999
```

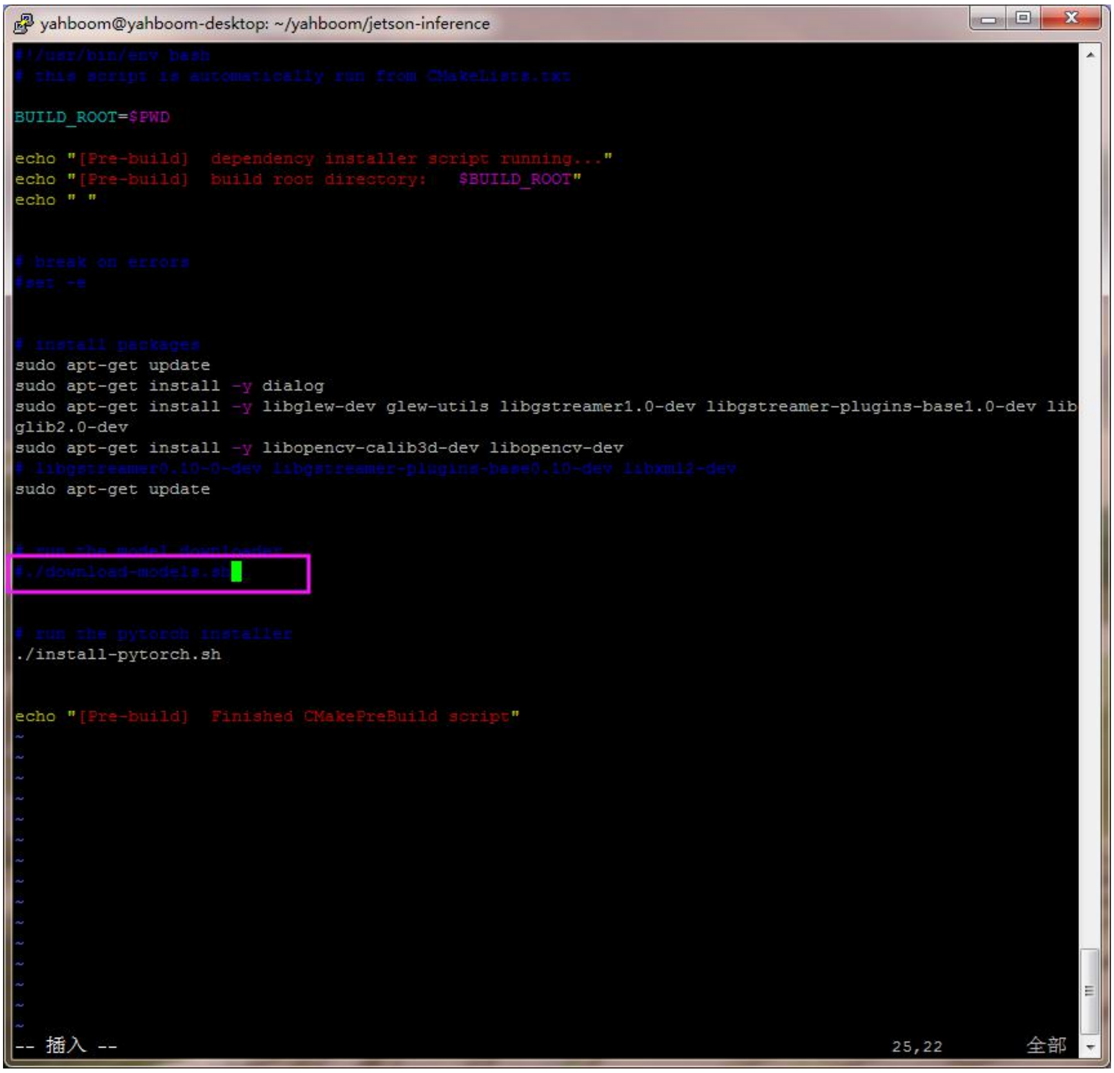
If it still does not work, it is a network problem in my own experience, and it is recommended to do it when the network condition is better.

```
cd jetson-inference
git submodule update --init
```

```
mkdir build        # Create build folder
cd build           # Enter build
cmake ../          # Run cmake, it will automatically execute CMakePrebuild.sh in the upper directory
```

!!!Note:
During this period, the system may prompt whether you need to install pytorch, please choose according to your personal needs.

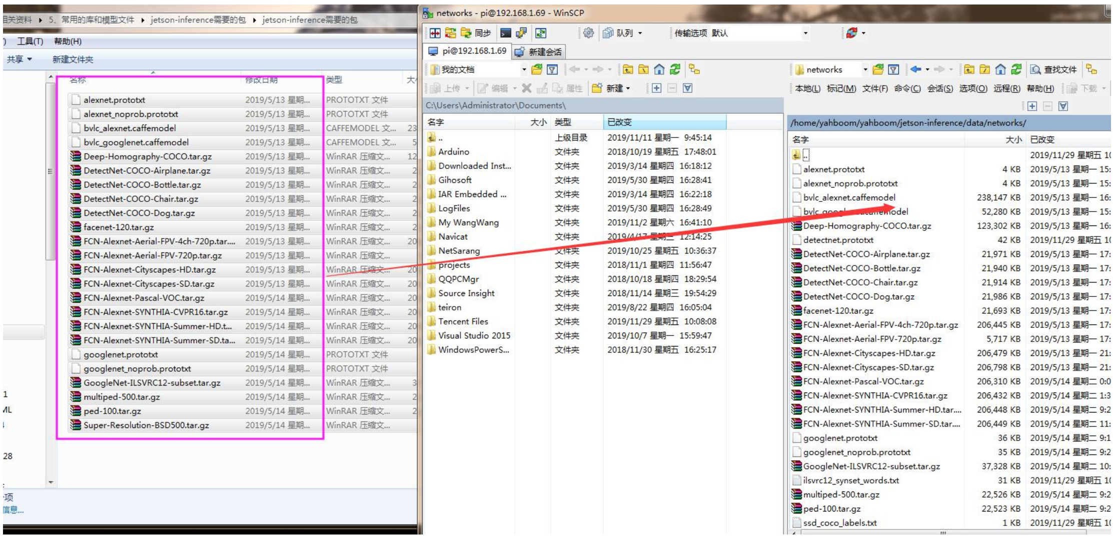
Proceed as follows:
Edit jetson-inference/CMakePrebuild.sh.
1)Comment out ./download-models.sh (with a # comment in front) as shown in Figure 1).
2) Transfer the jetson-inference package to the data/networks directory through WinSCP. As shown in Figure 2.



yahboom@yahboom-desktop: ~/yahboom/jetson-inference

```
#!/usr/bin/env bash
# this script is automatically run from CMakeLists.txt

BUILD_ROOT=$PWD

echo "[Pre-build]  dependency installer script running..."
echo "[Pre-build]  build root directory:   $BUILD_ROOT"
echo " "


# break on errors
#set -e


# install packages
sudo apt-get update
sudo apt-get install -y dialog
sudo apt-get install -y libglew-dev glew-utils libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev lib
glib2.0-dev
sudo apt-get install -y libopencv-calib3d-dev libopencv-dev
# libgstreamer0.10-0-dev libgstreamer-plugins-base0.10-dev libxml2-dev
sudo apt-get update


# run the model downloader
#./download-models.sh


# run the pytorch installer
./install-pytorch.sh


echo "[Pre-build]  Finished CMakePreBuild script"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- 插入 --                                                                    25,22         全部
```

Figure 1



相关资料 ▸ 5、常用的库和模型文件 ▸ jetson-inference需要的包 ▸ jetson-inference需要的包

名称	修改日期	类型	大小
alexnet.prototxt	2019/5/13 星期...	PROTOTXT 文件	
alexnet_noprob.prototxt	2019/5/13 星期...	PROTOTXT 文件	
bvlc_alexnet.caffemodel	2019/5/13 星期...	CAFFEMODEL 文...	23
bvlc_googlenet.caffemodel	2019/5/13 星期...	CAFFEMODEL 文...	5
Deep-Homography-COCO.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	12
DetectNet-COCO-Airplane.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	2
DetectNet-COCO-Bottle.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	2
DetectNet-COCO-Chair.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	2
DetectNet-COCO-Dog.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	2
facenet-120.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	2
FCN-Alexnet-Aerial-FPV-4ch-720p.tar....	2019/5/13 星期...	WinRAR 压缩文...	20
FCN-Alexnet-Aerial-FPV-720p.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	
FCN-Alexnet-Cityscapes-HD.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	20
FCN-Alexnet-Cityscapes-SD.tar.gz	2019/5/13 星期...	WinRAR 压缩文...	20
FCN-Alexnet-Pascal-VOC.tar.gz	2019/5/14 星期...	WinRAR 压缩文...	20
FCN-Alexnet-SYNTHIA-CVPR16.tar.gz	2019/5/14 星期...	WinRAR 压缩文...	20
FCN-Alexnet-SYNTHIA-Summer-HD.t...	2019/5/14 星期...	WinRAR 压缩文...	20
FCN-Alexnet-SYNTHIA-Summer-SD.ta...	2019/5/14 星期...	WinRAR 压缩文...	20
googlenet.prototxt	2019/5/14 星期...	PROTOTXT 文件	
googlenet_noprob.prototxt	2019/5/14 星期...	PROTOTXT 文件	
GoogleNet-ILSVRC12-subset.tar.gz	2019/5/14 星期...	WinRAR 压缩文...	3
multiped-500.tar.gz	2019/5/14 星期...	WinRAR 压缩文...	2
ped-100.tar.gz	2019/5/14 星期...	WinRAR 压缩文...	2
Super-Resolution-BSD500.tar.gz	2019/5/14 星期...	WinRAR 压缩文...	

networks - pi@192.168.1.69 - WinSCP

C:\Users\Administrator\Documents\

名字	大小	类型	已改变
..		上级目录	2019/11/11 星期一 9:45:14
Arduino		文件夹	2018/10/19 星期五 17:48:01
Downloaded Inst...		文件夹	2019/3/14 星期四 16:18:12
Gihosoft		文件夹	2019/5/30 星期四 16:28:41
IAR Embedded ...		文件夹	2019/3/14 星期四 16:22:18
LogFiles		文件夹	2019/5/30 星期四 16:28:49
My WangWang		文件夹	2019/11/2 星期六 16:41:10
Navicat		文件夹	2019/4/17 星期三 12:14:25
NetSarang		文件夹	2019/10/25 星期五 10:36:37
projects		文件夹	2018/11/1 星期四 11:56:47
QQPCMgr		文件夹	2018/10/18 星期四 18:29:54
Source Insight		文件夹	2018/11/14 星期三 19:54:29
teiron		文件夹	2019/8/22 星期四 16:05:04
Tencent Files		文件夹	2019/11/29 星期五 10:08:08
Visual Studio 2015		文件夹	2019/10/7 星期一 15:59:47
WindowsPowerS...		文件夹	2018/11/30 星期五 16:25:17

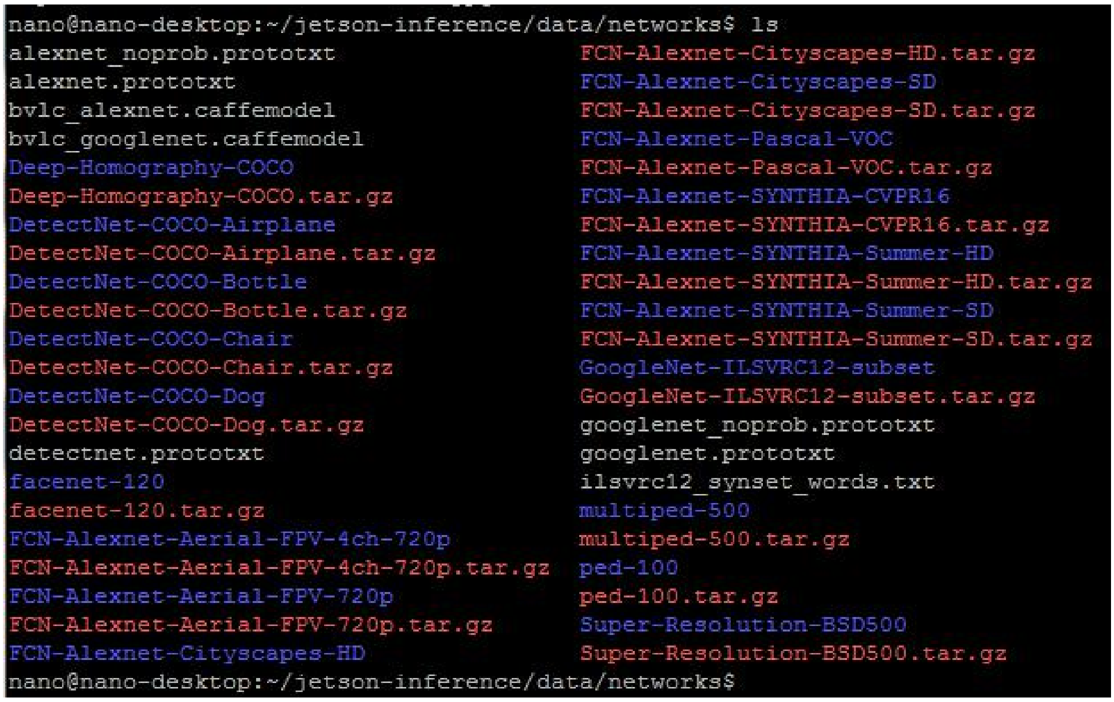
/home/yahboom/yahboom/jetson-inference/data/networks/

名字	大小	已改变
..		2019/11/29 星期五 1(
alexnet.prototxt	4 KB	2019/5/13 星期一 15:
alexnet_noprob.prototxt	4 KB	2019/5/13 星期一 15:
bvlc_alexnet.caffemodel	238,147 KB	2019/5/13 星期一 16:
bvlc_googlenet.caffemodel	52,280 KB	2019/5/13 星期一 15:
Deep-Homography-COCO.tar.gz	123,302 KB	2019/5/13 星期一 16:
detectnet.prototxt	42 KB	2019/11/29 星期五 1(
DetectNet-COCO-Airplane.tar.gz	21,971 KB	2019/5/13 星期一 17:
DetectNet-COCO-Bottle.tar.gz	21,940 KB	2019/5/13 星期一 17:
DetectNet-COCO-Chair.tar.gz	21,914 KB	2019/5/13 星期一 17:
DetectNet-COCO-Dog.tar.gz	21,986 KB	2019/5/13 星期一 17:
facenet-120.tar.gz	21,693 KB	2019/5/13 星期一 17:
FCN-Alexnet-Aerial-FPV-4ch-720p.tar.gz	206,445 KB	2019/5/13 星期一 17:
FCN-Alexnet-Aerial-FPV-720p.tar.gz	5,717 KB	2019/5/13 星期一 17:
FCN-Alexnet-Cityscapes-HD.tar.gz	206,479 KB	2019/5/13 星期一 21:
FCN-Alexnet-Cityscapes-SD.tar.gz	206,798 KB	2019/5/13 星期一 21:
FCN-Alexnet-Pascal-VOC.tar.gz	206,310 KB	2019/5/14 星期二 0:0
FCN-Alexnet-SYNTHIA-CVPR16.tar.gz	206,432 KB	2019/5/14 星期二 1:3
FCN-Alexnet-SYNTHIA-Summer-HD.tar....	206,448 KB	2019/5/14 星期二 9:2
FCN-Alexnet-SYNTHIA-Summer-SD.tar....	206,449 KB	2019/5/14 星期二 11:
googlenet.prototxt	36 KB	2019/5/14 星期二 9:1
googlenet_noprob.prototxt	35 KB	2019/5/14 星期二 9:2
GoogleNet-ILSVRC12-subset.tar.gz	37,328 KB	2019/5/14 星期二 10:
ilsvrc12_synset_words.txt	31 KB	2019/11/29 星期五 1(
multiped-500.tar.gz	22,526 KB	2019/5/14 星期二 9:2
ped-100.tar.gz	22,523 KB	2019/5/14 星期二 9:2
ssd_coco_labels.txt	1 KB	2019/11/29 星期五 1(

Figure 2

Input the following command to unzip.tar.gz files:

for tar in *.tar.gz; do tar xvf $tar; done



```
nano@nano-desktop:~/jetson-inference/data/networks$ ls
alexnet_noprob.prototxt                 FCN-Alexnet-Cityscapes-HD.tar.gz
alexnet.prototxt                        FCN-Alexnet-Cityscapes-SD
bvlc_alexnet.caffemodel                 FCN-Alexnet-Cityscapes-SD.tar.gz
bvlc_googlenet.caffemodel               FCN-Alexnet-Pascal-VOC
Deep-Homography-COCO                    FCN-Alexnet-Pascal-VOC.tar.gz
Deep-Homography-COCO.tar.gz             FCN-Alexnet-SYNTHIA-CVPR16
DetectNet-COCO-Airplane                 FCN-Alexnet-SYNTHIA-CVPR16.tar.gz
DetectNet-COCO-Airplane.tar.gz          FCN-Alexnet-SYNTHIA-Summer-HD
DetectNet-COCO-Bottle                   FCN-Alexnet-SYNTHIA-Summer-HD.tar.gz
DetectNet-COCO-Bottle.tar.gz            FCN-Alexnet-SYNTHIA-Summer-SD
DetectNet-COCO-Chair                    FCN-Alexnet-SYNTHIA-Summer-SD.tar.gz
DetectNet-COCO-Chair.tar.gz             GoogleNet-ILSVRC12-subset
DetectNet-COCO-Dog                      GoogleNet-ILSVRC12-subset.tar.gz
DetectNet-COCO-Dog.tar.gz               googlenet_noprob.prototxt
detectnet.prototxt                      googlenet.prototxt
facenet-120                             ilsvrc12_synset_words.txt
facenet-120.tar.gz                      multiped-500
FCN-Alexnet-Aerial-FPV-4ch-720p         multiped-500.tar.gz
FCN-Alexnet-Aerial-FPV-4ch-720p.tar.gz  ped-100
FCN-Alexnet-Aerial-FPV-720p             ped-100.tar.gz
FCN-Alexnet-Aerial-FPV-720p.tar.gz      Super-Resolution-BSD500
FCN-Alexnet-Cityscapes-HD               Super-Resolution-BSD500.tar.gz
nano@nano-desktop:~/jetson-inference/data/networks$
```

Input the following command to unzip.tar.gz files:

for gz in *.gz; do gunzip $gz; done

Input the following command to unzip multiple .tar.gz files:

for tar in *.tar.gz;   do tar xvf $tar; done

After cmake is successful, we need to compile

cd jetson-inference/build

# Make here without sudo

# Back -j4 uses 4 CPU cores to compile at the same time, reducing time

make (or make -j4)   Note: (In the build directory)

sudo make install   Note: (In the build directory)

If the compilation is successful, the following folder structure will be generated.

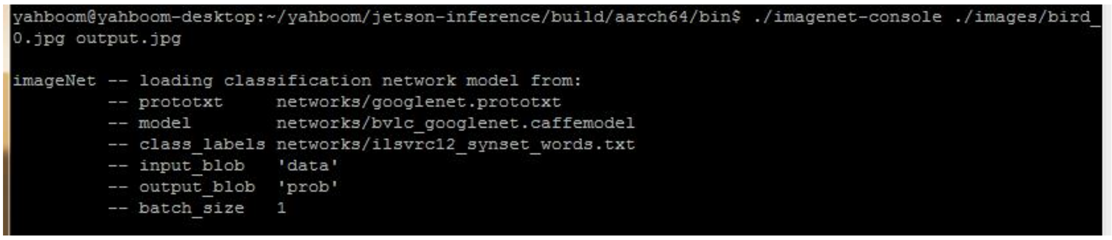
```
|-build
    \aarch64          (64-bit)
        \bin          where the sample binaries are built to
        \include      where the headers reside
        \lib          where the libraries are build to
    \armhf              (32-bit)
        \bin          where the sample binaries are built to
        \include      where the headers reside
        \lib          where the libraries are build to
```

3.Testing

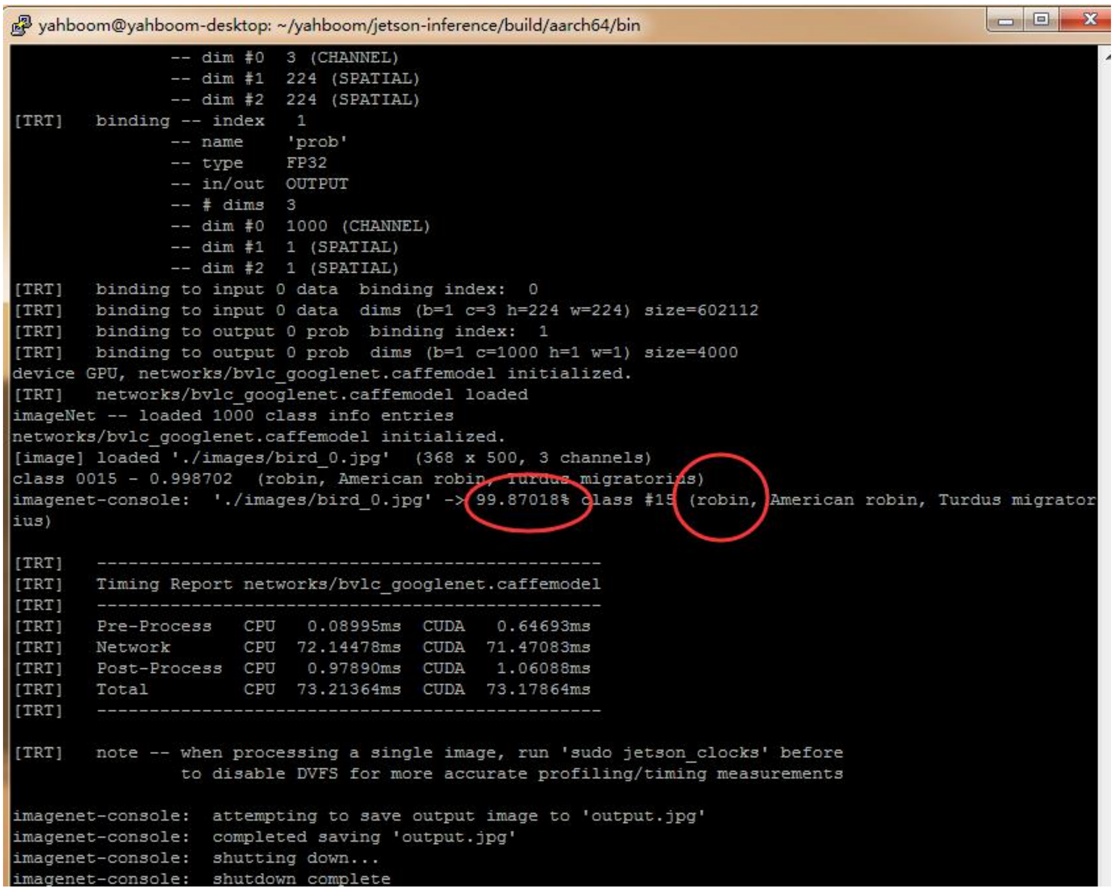
We need to input the following command:

```
cd jetson-inference/build/aarch64/bin
./imagenet-console orange_0.jpg output_0.jpg
```

After waiting patiently, we will see the interface shown below.
(The first execution will take a long time)



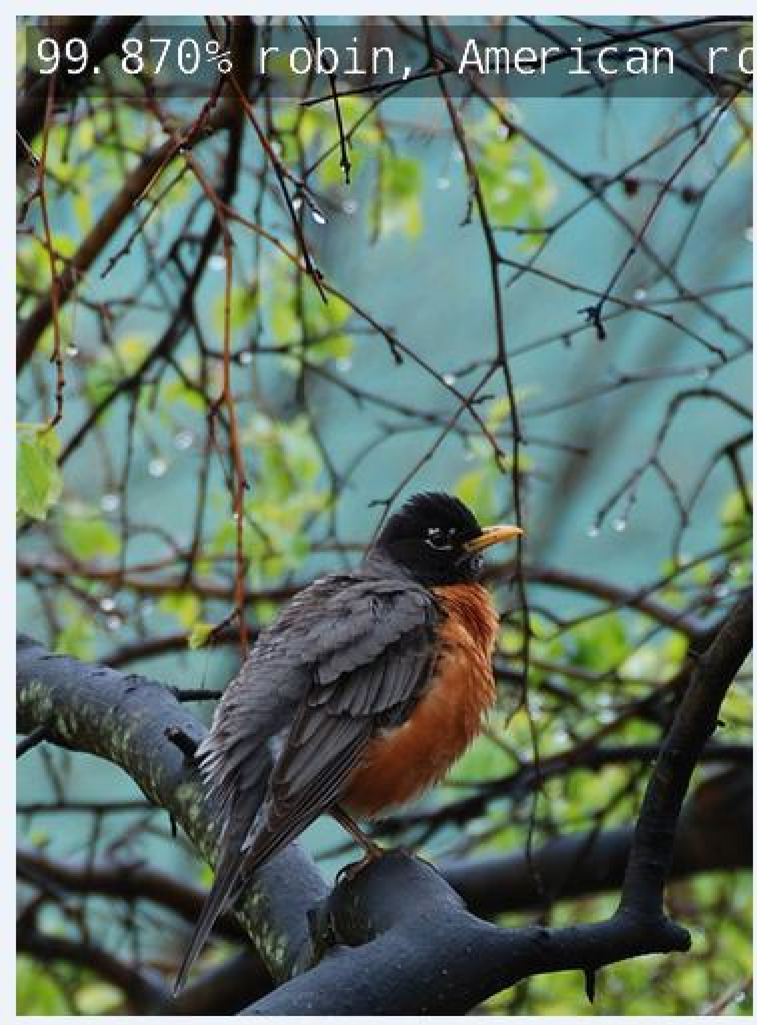
```
yahboom@yahboom-desktop:~/yahboom/jetson-inference/build/aarch64/bin$ ./imagenet-console ./images/bird_
0.jpg output.jpg

imageNet -- loading classification network model from:
         -- prototxt     networks/googlenet.prototxt
         -- model        networks/bvlc_googlenet.caffemodel
         -- class_labels networks/ilsvrc12_synset_words.txt
         -- input_blob   'data'
         -- output_blob  'prob'
         -- batch_size   1
```



yahboom@yahboom-desktop: ~/yahboom/jetson-inference/build/aarch64/bin

```
               -- dim #0  3 (CHANNEL)
               -- dim #1  224 (SPATIAL)
               -- dim #2  224 (SPATIAL)
[TRT]   binding -- index   1
               -- name    'prob'
               -- type    FP32
               -- in/out  OUTPUT
               -- # dims  3
               -- dim #0  1000 (CHANNEL)
               -- dim #1  1 (SPATIAL)
               -- dim #2  1 (SPATIAL)
[TRT]   binding to input 0 data  binding index:  0
[TRT]   binding to input 0 data  dims (b=1 c=3 h=224 w=224) size=602112
[TRT]   binding to output 0 prob  binding index:  1
[TRT]   binding to output 0 prob  dims (b=1 c=1000 h=1 w=1) size=4000
device GPU, networks/bvlc_googlenet.caffemodel initialized.
[TRT]   networks/bvlc_googlenet.caffemodel loaded
imageNet -- loaded 1000 class info entries
networks/bvlc_googlenet.caffemodel initialized.
[image] loaded './images/bird_0.jpg'  (368 x 500, 3 channels)
class 0015 - 0.998702  (robin, American robin, Turdus migratorius)
imagenet-console:  './images/bird_0.jpg' -> 99.87018% class #15 (robin, American robin, Turdus migrator
ius)

[TRT]   ----------------------------------------------
[TRT]   Timing Report networks/bvlc_googlenet.caffemodel
[TRT]   ----------------------------------------------
[TRT]   Pre-Process   CPU   0.08995ms  CUDA   0.64693ms
[TRT]   Network       CPU  72.14478ms  CUDA  71.47083ms
[TRT]   Post-Process  CPU   0.97890ms  CUDA   1.06088ms
[TRT]   Total         CPU  73.21364ms  CUDA  73.17864ms
[TRT]   ----------------------------------------------

[TRT]   note -- when processing a single image, run 'sudo jetson_clocks' before
                to disable DVFS for more accurate profiling/timing measurements

imagenet-console:  attempting to save output image to 'output.jpg'
imagenet-console:  completed saving 'output.jpg'
imagenet-console:  shutting down...
imagenet-console:  shutdown complete
```

We need to find the corresponding directory to view **output_0.jpg**, the recognition result will be displayed at the top of the picture, as shown below.



For more detail, please see the official documentation:
Official Demo:
https://developer.nvidia.com/embedded/twodaystoademo
Official TensorRT tutorial:
https://docs.nvidia.com/deeplearning/sdk/tensorrt-sample-support-guide/index.html