基于duo开发板的googlenet图像分类

1.配置docker开发环境

```
sudo apt install docker.io
sudo systemctl start docker
sudo systemctl enable docker
sudo groupadd docker
sudo usermod -aG docker $USER
newgrp docker
```

启动容器并获取开发工具包

```
docker pull sophgo/tpuc_dev:v2.2
docker run --privileged --name googlenet -v /workspace -it sophgo/tpuc_dev:v2.2
apt-get update
apt-get install net-tools
wget --user='cvitek_mlir_2023' --password='7&2Wd%cu5k' ftp://218.17.249.213/home/tpu-mlir_v1.2.89-g77a2268f-20230703.tar.gz
```

容器名默认用googlenet了，有需要可以自行更改。

将工具包解压并添加环境变量

```
tar -zxvf tpu-mlir_v1.2.89-g77a2268f-20230703.tar.gz
source ./tpu-mlir_v1.2.89-g77a2268f-20230703/envsetup.sh
```

2.在docker中准备工作目录

建立`googlenet`工作目录，注意是与`tpu-mlir_xxxx`同级的目录，并将模型文件和图片文件都放入该目录下

```
mkdir googlenet && cd googlenet
wget https://github.com/onnx/models/raw/main/vision/classification/inception_and_googlenet/googlenet/model/googlenet-12.onnx
cp -rf $TPUC_ROOT/regression/dataset/ILSVRC2012 .
cp -rf $TPUC_ROOT/regression/image .
```

这里的`$TPUC_ROOT`是环境变量，对应`tpu-mlir_xxxx`目录

然后在当前目录下新建`work`目录

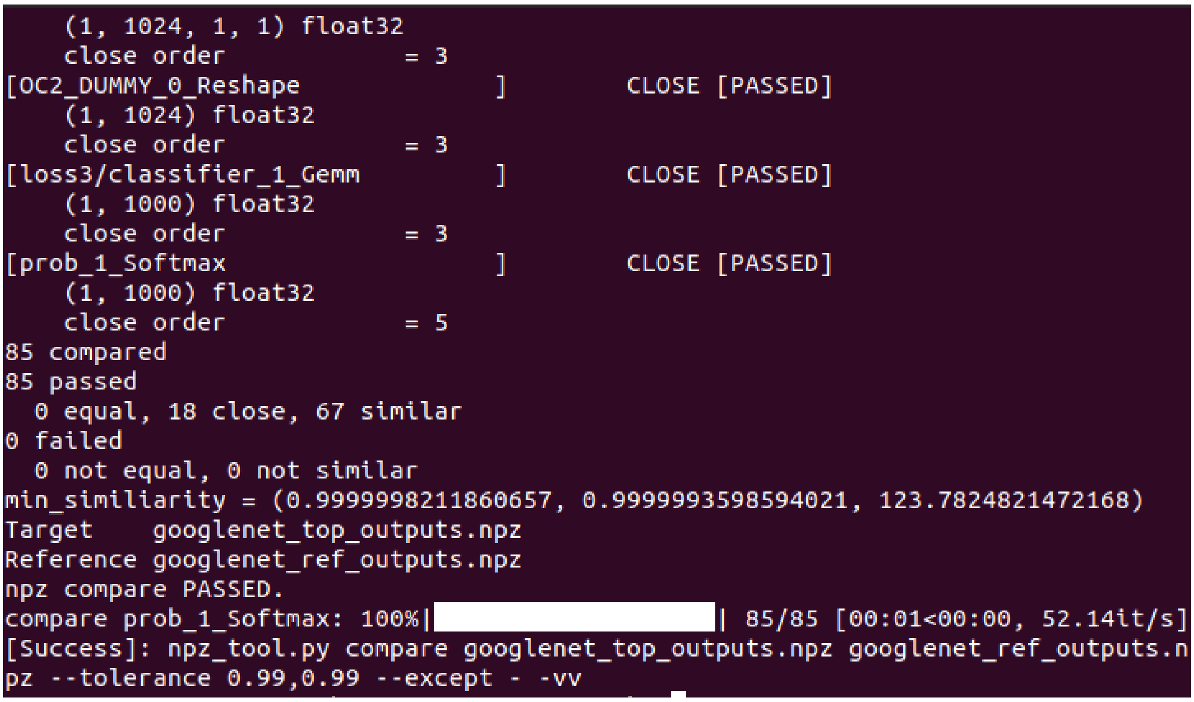
```
mkdir work && cd work
```

3.ONNX转MLIR

本例中，模型是RGB输入，`mean`和`scale`分别为`123.675,116.28,103.53`和`0.0171,0.0175,0.0174`
将onnx模型转换为mlir模型的命令如下：

```
model_transform.py \
  --model_name googlenet \
  --model_def ../googlenet-12.onnx \
  --test_input ../image/cat.jpg \
  --input_shapes [[1,3,224,224]] \
  --resize_dims 256,256 \
  --mean 123.675,116.28,103.53 \
  --scale 0.0171,0.0175,0.0174 \
  --pixel_format rgb \
  --test_result googlenet_top_outputs.npz \
  --mlir googlenet.mlir
```

运行成功示例：



```
    (1, 1024, 1, 1) float32
    close order            = 3
[OC2_DUMMY_0_Reshape                ]        CLOSE [PASSED]
    (1, 1024) float32
    close order            = 3
[loss3/classifier_1_Gemm            ]        CLOSE [PASSED]
    (1, 1000) float32
    close order            = 3
[prob_1_Softmax                     ]        CLOSE [PASSED]
    (1, 1000) float32
    close order            = 5
85 compared
85 passed
  0 equal, 18 close, 67 similar
0 failed
  0 not equal, 0 not similar
min_similiarity = (0.9999998211860657, 0.9999993598594021, 123.7824821472168)
Target    googlenet_top_outputs.npz
Reference googlenet_ref_outputs.npz
npz compare PASSED.
compare prob_1_Softmax: 100%|███████████████████| 85/85 [00:01<00:00, 52.14it/s]
[Success]: npz_tool.py compare googlenet_top_outputs.npz googlenet_ref_outputs.n
pz --tolerance 0.99,0.99 --except - -vv
```

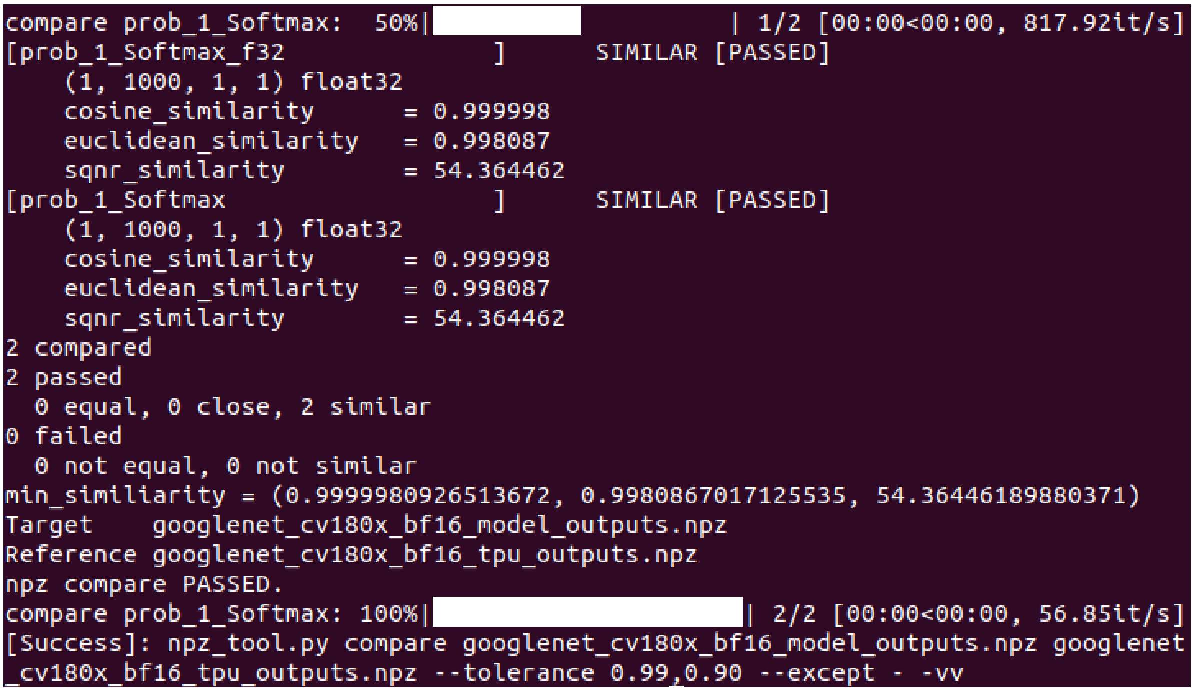
转成mlir模型后，会生成一个`googlenet.mlir`文件，该文件即为mlir模型文件，还会生成一个`googlenet_in_f32.npz`文件和一个`googlenet_top_outputs.npz`文件，该文件是后续转模型的输入文件

4.MLIR转BF16模型

将mlir模型转换为bf16模型的命令如下：

```
model_deploy.py \
  --mlir googlenet.mlir \
  --quantize BF16 \
  --chip cv180x \
  --test_input googlenet_in_f32.npz \
  --test_reference googlenet_top_outputs.npz \
  --model googlenet_cv180x_bf16.cvimodel
```

编译成功示例：



```
compare prob_1_Softmax:  50%|██████████          | 1/2 [00:00<00:00, 817.92it/s]
[prob_1_Softmax_f32             ]      SIMILAR [PASSED]
    (1, 1000, 1, 1) float32
    cosine_similarity      = 0.999998
    euclidean_similarity   = 0.998087
    sqnr_similarity        = 54.364462
[prob_1_Softmax                 ]      SIMILAR [PASSED]
    (1, 1000, 1, 1) float32
    cosine_similarity      = 0.999998
    euclidean_similarity   = 0.998087
    sqnr_similarity        = 54.364462
2 compared
2 passed
  0 equal, 0 close, 2 similar
0 failed
  0 not equal, 0 not similar
min_similiarity = (0.9999980926513672, 0.9980867017125535, 54.36446189880371)
Target    googlenet_cv180x_bf16_model_outputs.npz
Reference googlenet_cv180x_bf16_tpu_outputs.npz
npz compare PASSED.
compare prob_1_Softmax: 100%|█████████████████████| 2/2 [00:00<00:00, 56.85it/s]
[Success]: npz_tool.py compare googlenet_cv180x_bf16_model_outputs.npz googlenet
_cv180x_bf16_tpu_outputs.npz --tolerance 0.99,0.90 --except - -vv
```

编译完成后，会生成`googlenet_cv180x_bf16.cvimodel`文件

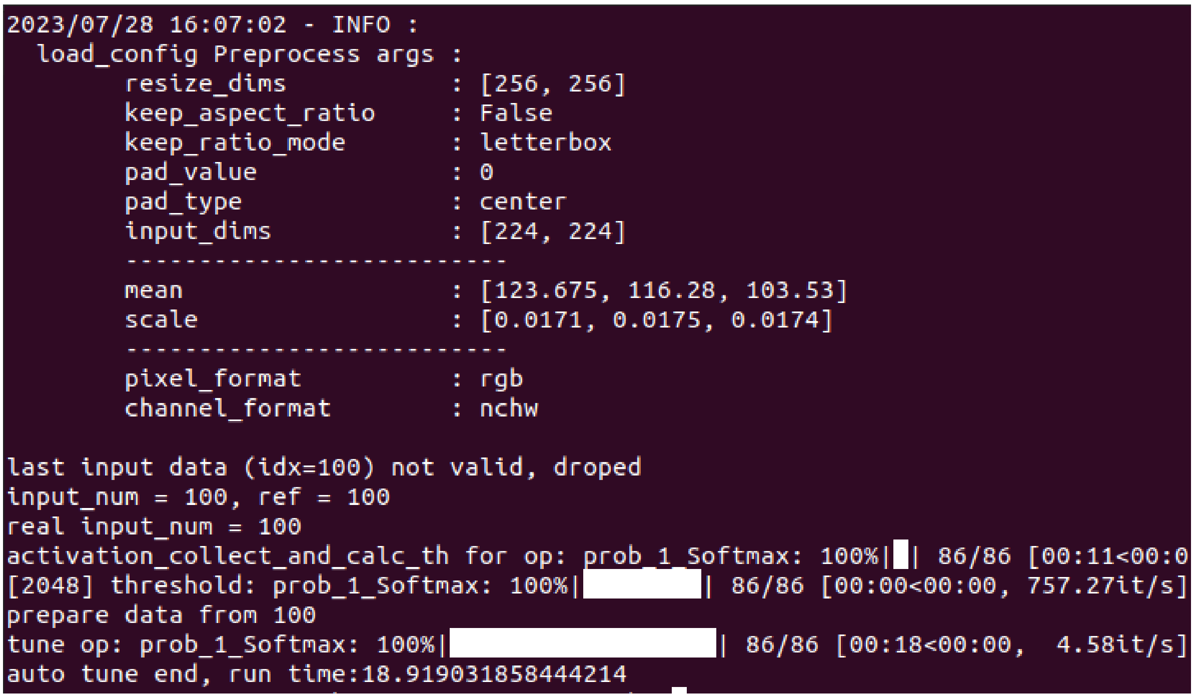
5.MLIR转INT8模型

生成校准表

在转int8模型之前需要先生成校准表，这里用现有的100张来自ILSVRC2012的图片举例，执行calibration：

```
run_calibration.py googlenet.mlir \
  --dataset ../ILSVRC2012 \
  --input_num 100 \
  -o googlenet_cali_table
```

运行成功示例：



```
2023/07/28 16:07:02 - INFO :
  load_config Preprocess args :
        resize_dims           : [256, 256]
        keep_aspect_ratio     : False
        keep_ratio_mode       : letterbox
        pad_value             : 0
        pad_type              : center
        input_dims            : [224, 224]
        --------------------------
        mean                  : [123.675, 116.28, 103.53]
        scale                 : [0.0171, 0.0175, 0.0174]
        --------------------------
        pixel_format          : rgb
        channel_format        : nchw

last input data (idx=100) not valid, droped
input_num = 100, ref = 100
real input_num = 100
activation_collect_and_calc_th for op: prob_1_Softmax: 100%|█| 86/86 [00:11<00:0
[2048] threshold: prob_1_Softmax: 100%|████████| 86/86 [00:00<00:00, 757.27it/s]
prepare data from 100
tune op: prob_1_Softmax: 100%|██████████████████| 86/86 [00:18<00:00,  4.58it/s]
auto tune end, run time:18.919031858444214
```

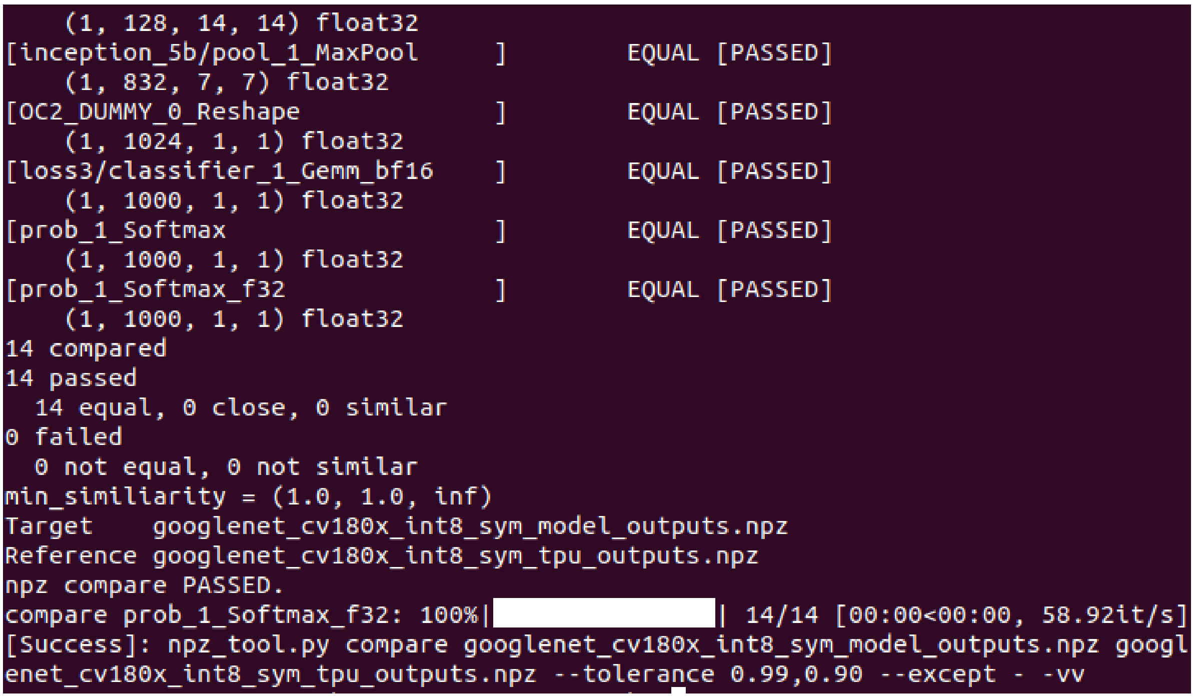
运行完成后，会生成`googlenet_cali_table`文件，该文件用于后续编译int8模型

编译为int8模型

将mlir模型转换为int8模型的命令如下：

```
model_deploy.py \
  --mlir googlenet.mlir \
  --quantize INT8 \
  --calibration_table googlenet_cali_table \
  --chip cv180x \
  --test_input ../image/cat.jpg \
  --test_reference googlenet_top_outputs.npz \
  --compare_all \
  --fuse_preprocess \
  --model googlenet_cv180x_int8_fuse.cvimodel
```

运行成功示例：



```
    (1, 128, 14, 14) float32
[inception_5b/pool_1_MaxPool     ]        EQUAL [PASSED]
    (1, 832, 7, 7) float32
[OC2_DUMMY_0_Reshape             ]        EQUAL [PASSED]
    (1, 1024, 1, 1) float32
[loss3/classifier_1_Gemm_bf16    ]        EQUAL [PASSED]
    (1, 1000, 1, 1) float32
[prob_1_Softmax                  ]        EQUAL [PASSED]
    (1, 1000, 1, 1) float32
[prob_1_Softmax_f32              ]        EQUAL [PASSED]
    (1, 1000, 1, 1) float32
14 compared
14 passed
  14 equal, 0 close, 0 similar
0 failed
  0 not equal, 0 not similar
min_similiarity = (1.0, 1.0, inf)
Target    googlenet_cv180x_int8_sym_model_outputs.npz
Reference googlenet_cv180x_int8_sym_tpu_outputs.npz
npz compare PASSED.
compare prob_1_Softmax_f32: 100%|███████████████| 14/14 [00:00<00:00, 58.92it/s]
[Success]: npz_tool.py compare googlenet_cv180x_int8_sym_model_outputs.npz googl
enet_cv180x_int8_sym_tpu_outputs.npz --tolerance 0.99,0.90 --except - -vv
```

编译完成后，会生成`googlenet_cv180x_int8_fuse.cvimodel`文件

6.在duo开发板上进行验证

连接duo开发板

根据教程完成duo开发板与电脑的连接，并使用`Mobaxterm`开启终端操作duo开发板

获取cvitek_tpu_sdk

可以从下载站台中获取开发工具包`cvitek_tpu_sdk_cv180x_musl_riscv64_rvv.tar.gz`，注意需要选择`cv180x`的工具包，下载站台如下：

```
wget --user='cvitek_mlir_2023' --password='7&2Wd%cu5k'
ftp://218.17.249.213/home/tpu_sdk_t4.1.0-14-
g3e77050/cvitek_tpu_sdk_cv180x_musl_riscv64_rvv.tar.gz
```

在docker中进行解压

```
tar -zxvf cvitek_tpu_sdk_cv180x_musl_riscv64_rvv.tar.gz
```

解压完成后会生成`cvitek_tpu_sdk`文件夹

将开发工具包和模型文件拷贝到开发板上

在duo开发板的终端中，新建文件目录`/home/milkv/`

```
mkdir /home/milkv && cd /home/milkv
```

在docker的终端中，将开发工具包和模型文件拷贝到开发板上

```
scp -r cvitek_tpu_sdk root@192.168.42.1:/home/milkv
scp /workspace/googlenet/work/googlenet_cv180x_bf16.cvimodel
root@192.168.42.1:/home/milkv/cvitek_tpu_sdk
scp /workspace/googlenet/work/googlenet_cv180x_int8_fuse.cvimodel
root@192.168.42.1:/home/milkv/cvitek_tpu_sdk
```

设置环境变量

在duo开发板的终端中，进行环境变量的设置

```
cd ./cvitek_tpu_sdk
source ./envs_tpu_sdk.sh
```

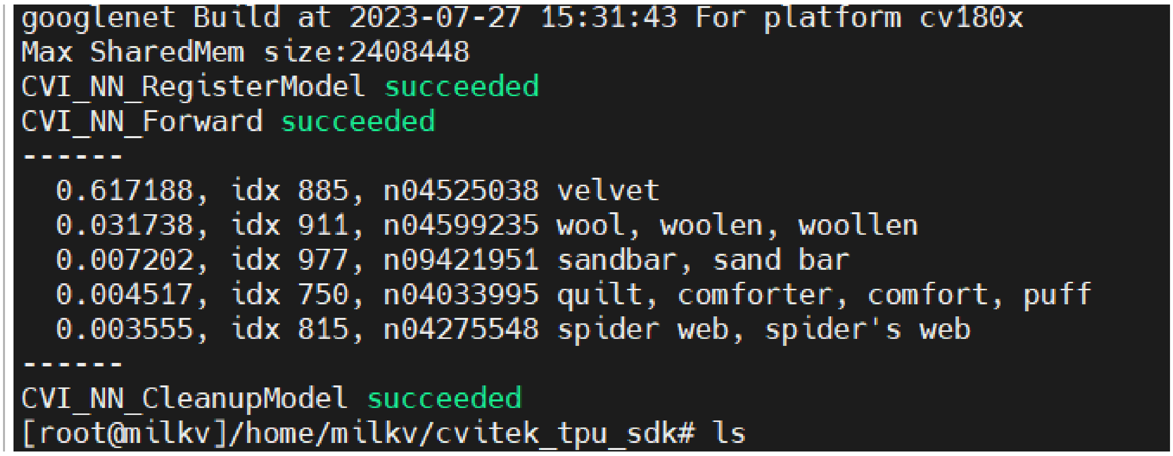
进行图像分类

在duo开发板中，对cat.jpg进行分类：

在duo开发板的终端中，输入如下命令，使用`googlenet_cv180x_bf16.cvimodel`模型进行图像分类：

```
./samples/bin/cvi_sample_classifier_bf16 \
  ./googlenet_cv180x_bf16.cvimodel \
  ./samples/data/cat.jpg \
  ./samples/data/synset_words.txt
```

运行成功后会输出如下信息：

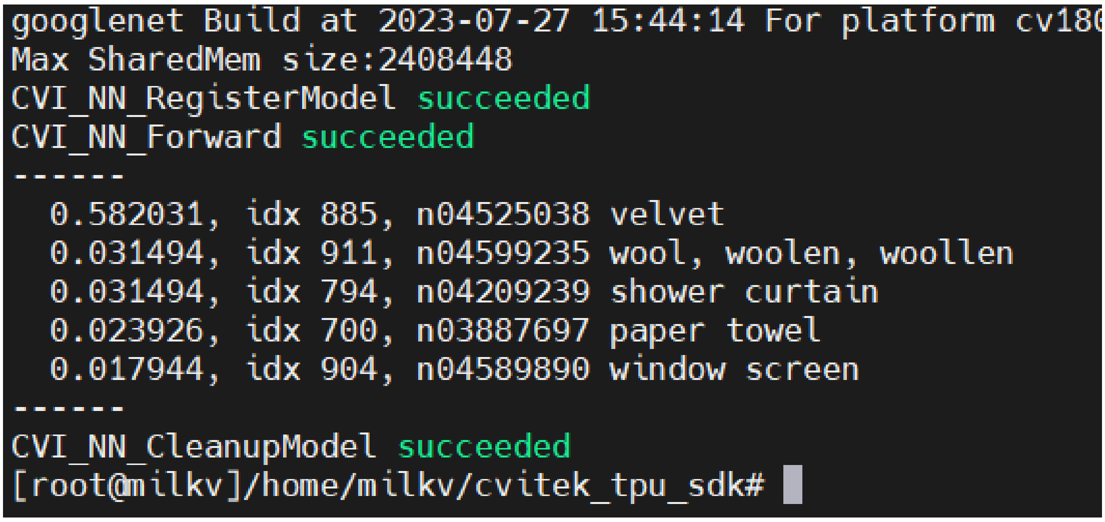


```
googlenet Build at 2023-07-27 15:31:43 For platform cv180x
Max SharedMem size:2408448
CVI_NN_RegisterModel succeeded
CVI_NN_Forward succeeded
------
  0.617188, idx 885, n04525038 velvet
  0.031738, idx 911, n04599235 wool, woolen, woollen
  0.007202, idx 977, n09421951 sandbar, sand bar
  0.004517, idx 750, n04033995 quilt, comforter, comfort, puff
  0.003555, idx 815, n04275548 spider web, spider's web
------
CVI_NN_CleanupModel succeeded
[root@milkv]/home/milkv/cvitek_tpu_sdk# ls
```

在duo开发板的终端中，输入如下命令，使用`googlenet_cv180x_int8_fuse.cvimodel`模型进行图像分类：

```
./samples/bin/cvi_sample_classifier_fused_preprocess \
  ./googlenet_cv180x_int8_fuse.cvimodel \
  ./samples/data/cat.jpg \
  ./samples/data/synset_words.txt
```

运行成功后会输出如下信息：



```
googlenet Build at 2023-07-27 15:44:14 For platform cv18
Max SharedMem size:2408448
CVI_NN_RegisterModel succeeded
CVI_NN_Forward succeeded
------
  0.582031, idx 885, n04525038 velvet
  0.031494, idx 911, n04599235 wool, woolen, woollen
  0.031494, idx 794, n04209239 shower curtain
  0.023926, idx 700, n03887697 paper towel
  0.017944, idx 904, n04589890 window screen
------
CVI_NN_CleanupModel succeeded
[root@milkv]/home/milkv/cvitek_tpu_sdk#
```