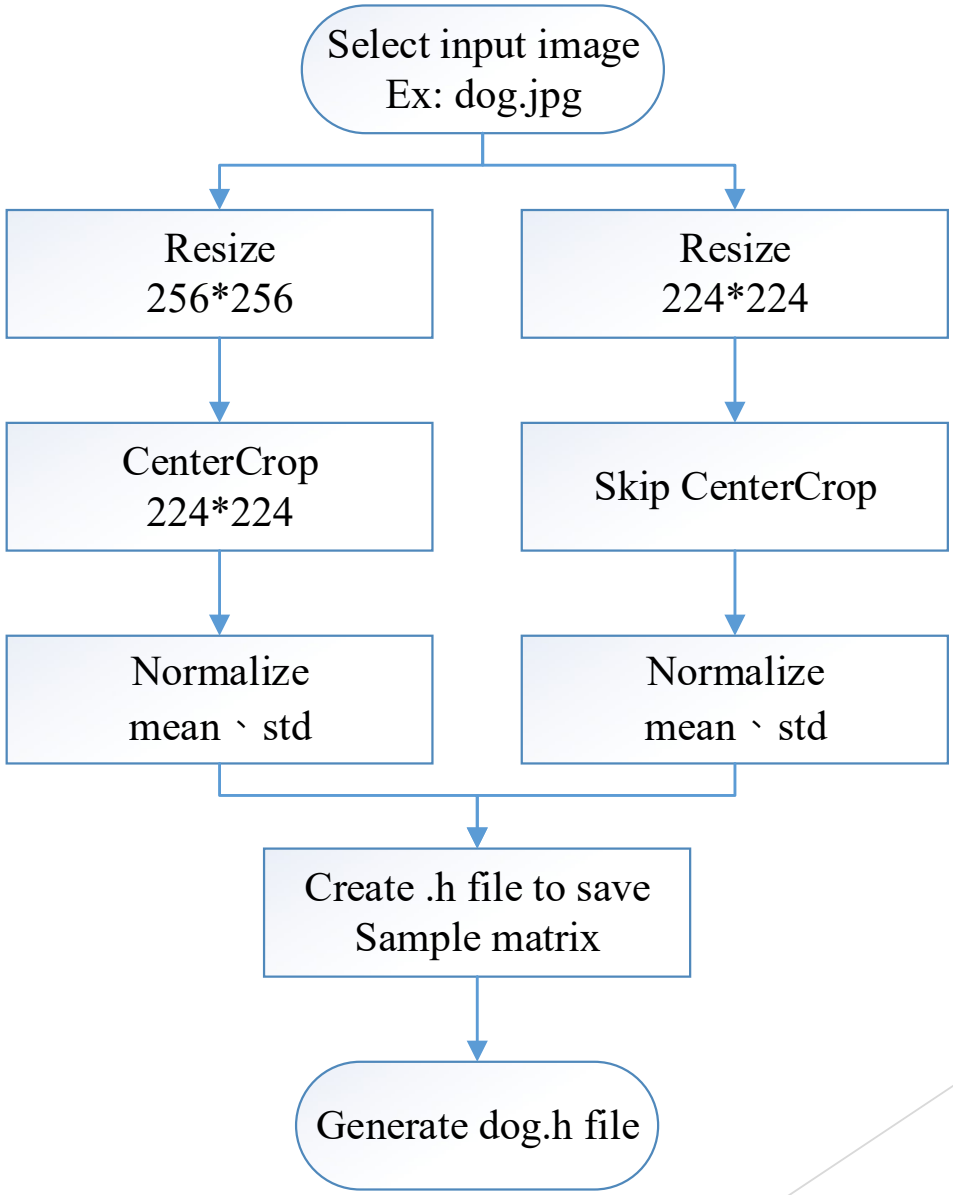
如何輸入圖片與正規化方式

作者:陳眾科

Outline

- 流程圖說明
- 圖片正規化方式
- 撰寫Makefile完成編譯host文件
- 實際操作
- 下周研究方向

流程圖說明


Select input image
Ex: dog.jpg
Resize
256*256
Resize
224*224
CenterCrop
224*224
Skip CenterCrop
Normalize
mean、std
Normalize
mean、std
Create .h file to save
Sample matrix
Generate dog.h file

正規化方式

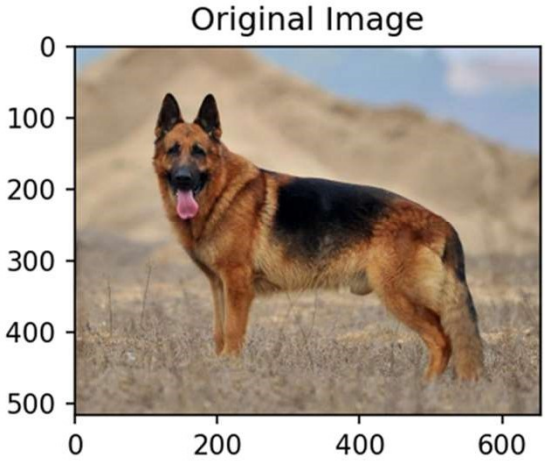
引入一些模組、輸入待處理的圖片

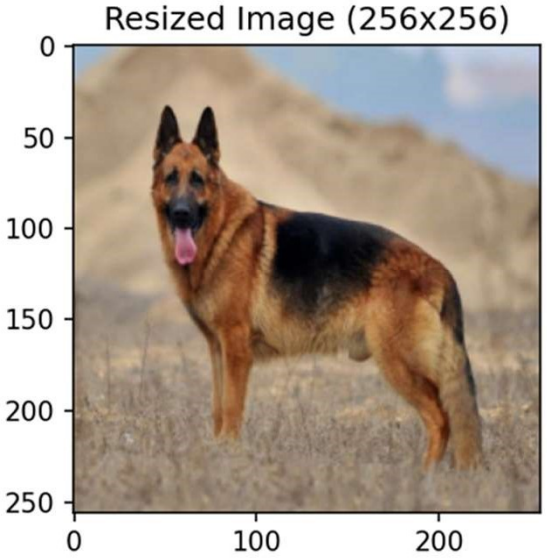
```python
from PIL import Image
import numpy as np
import torchvision.transforms as transforms
import torch

# 加载JPEG图像
image_path = r'D:\learn_pytorch\dog.jpg'  # 图像文件的完整路径
image = Image.open(image_path)
```

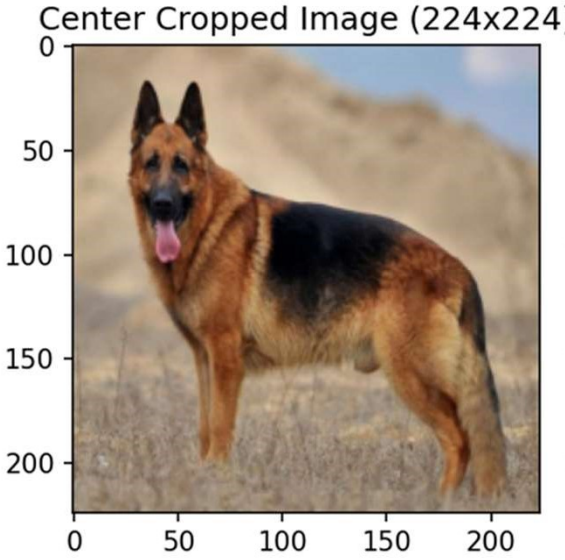
正規化過程

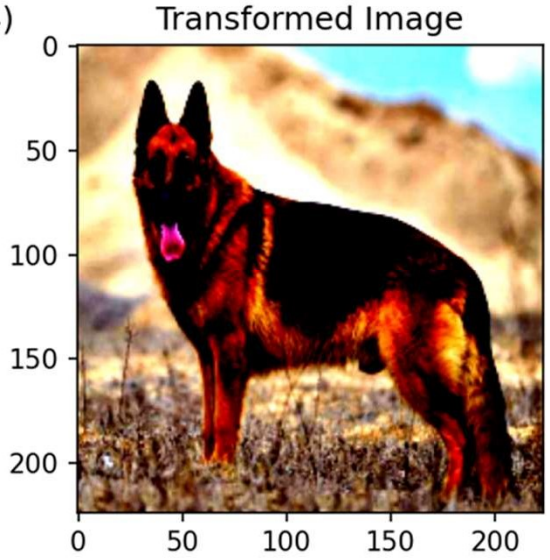
```python
# 定义transforms，包括resize、center crop和标准化
transform = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
# 应用transforms
image = transform(image)
```

過程展示

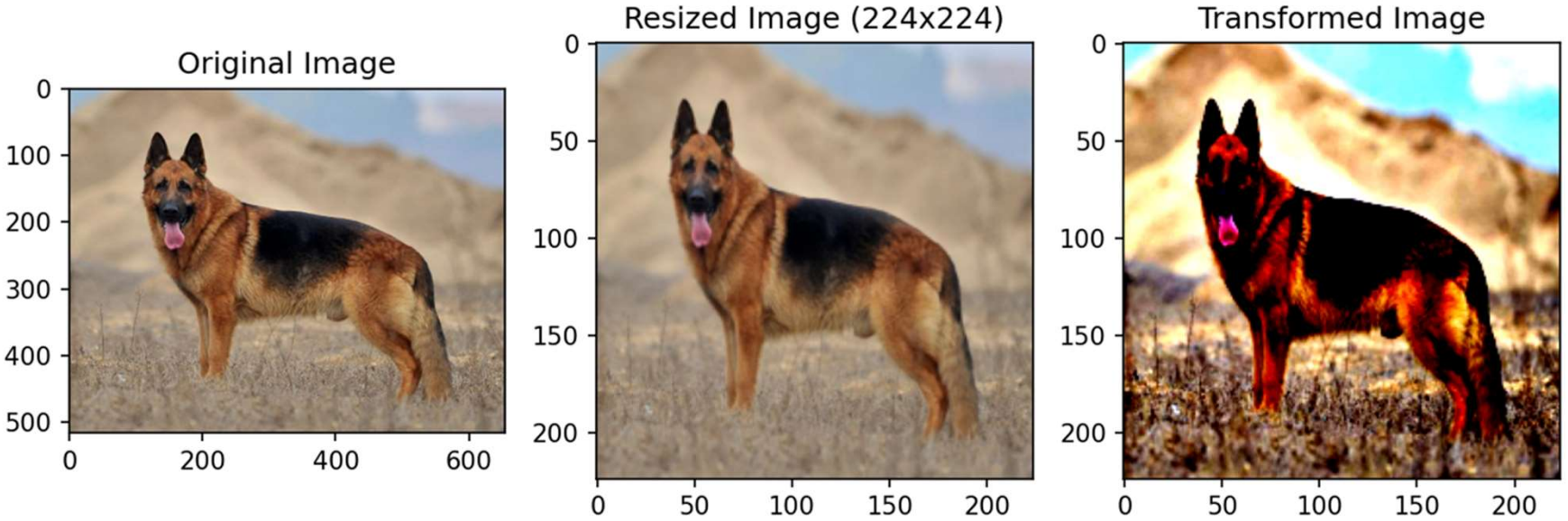

Original Image
0
100
200
300
400
500
0
200
400
600


Resized Image (256x256)
0
50
100
150
200
250
0
100
200


Center Cropped Image (224x224)
0
50
100
150
200
0
50
100
150
200

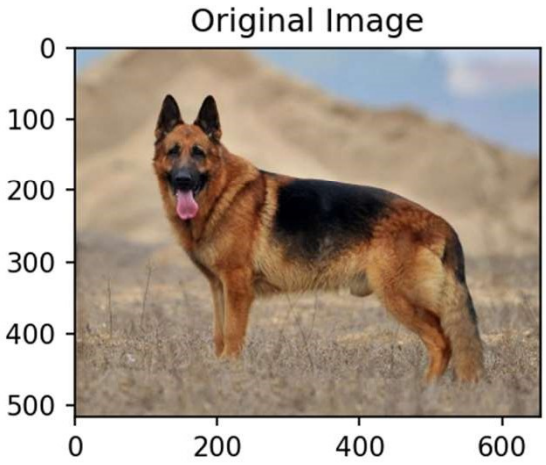

Transformed Image
0
50
100
150
200
0
50
100
150
200

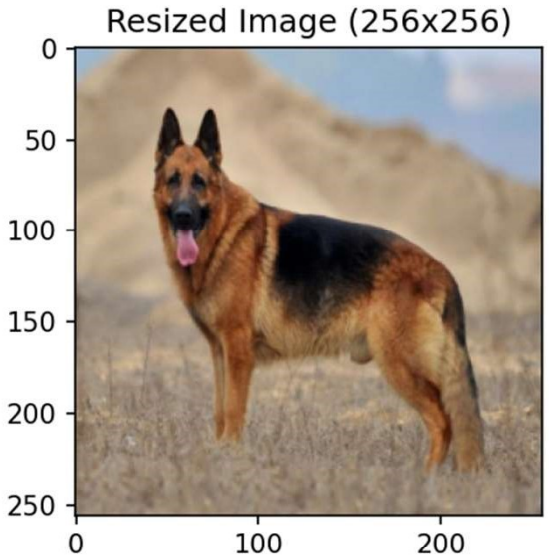
如果直接Resize變成224*224?

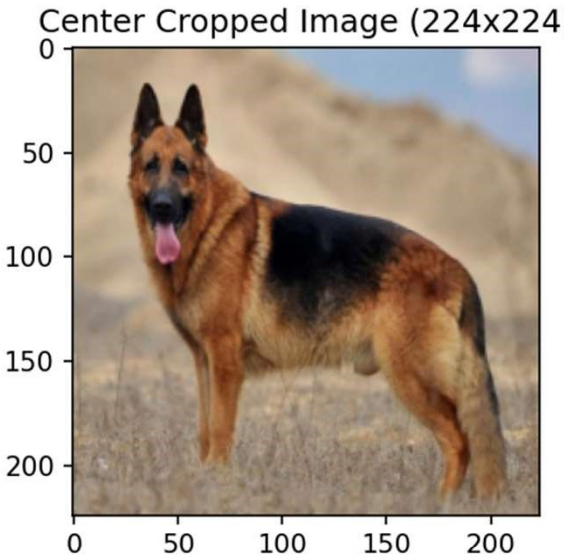


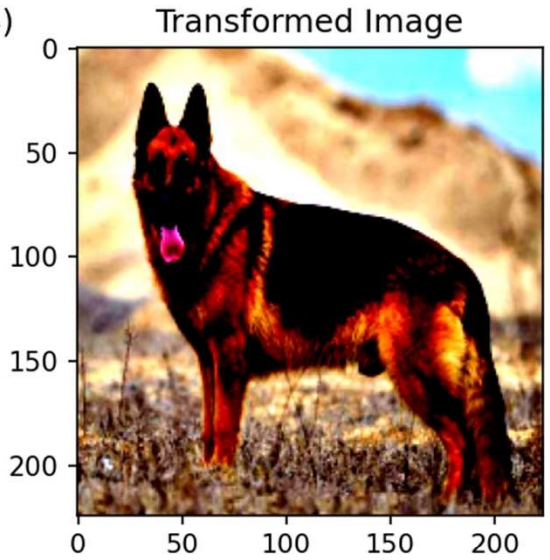

```python
# 定义transforms，包括resize、center crop和标准化
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    #transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
```

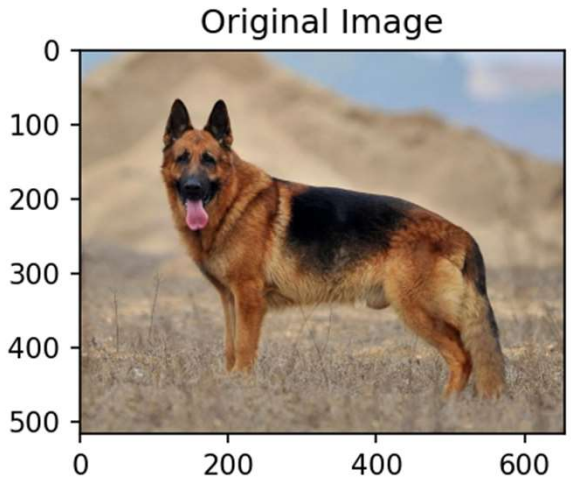
兩種方法比較

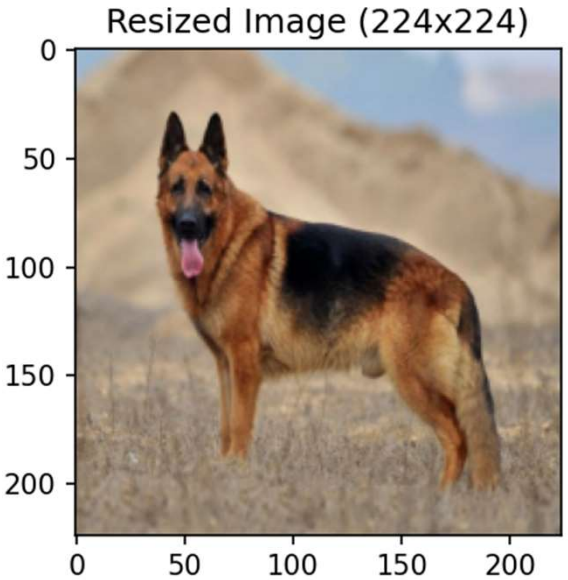

Original Image

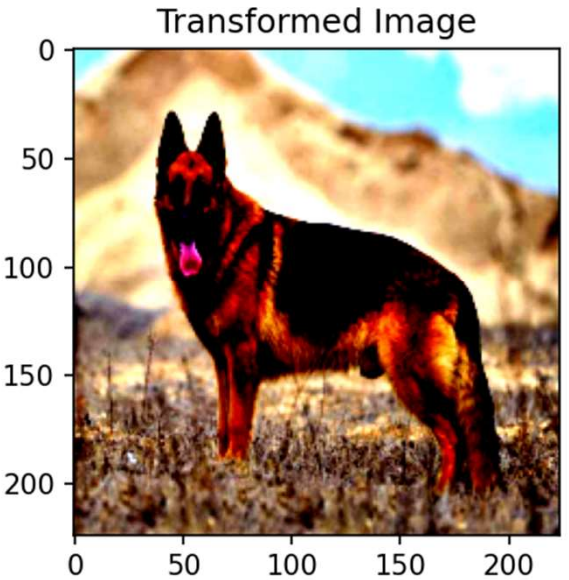

Resized Image (256x256)


Center Cropped Image (224x224)

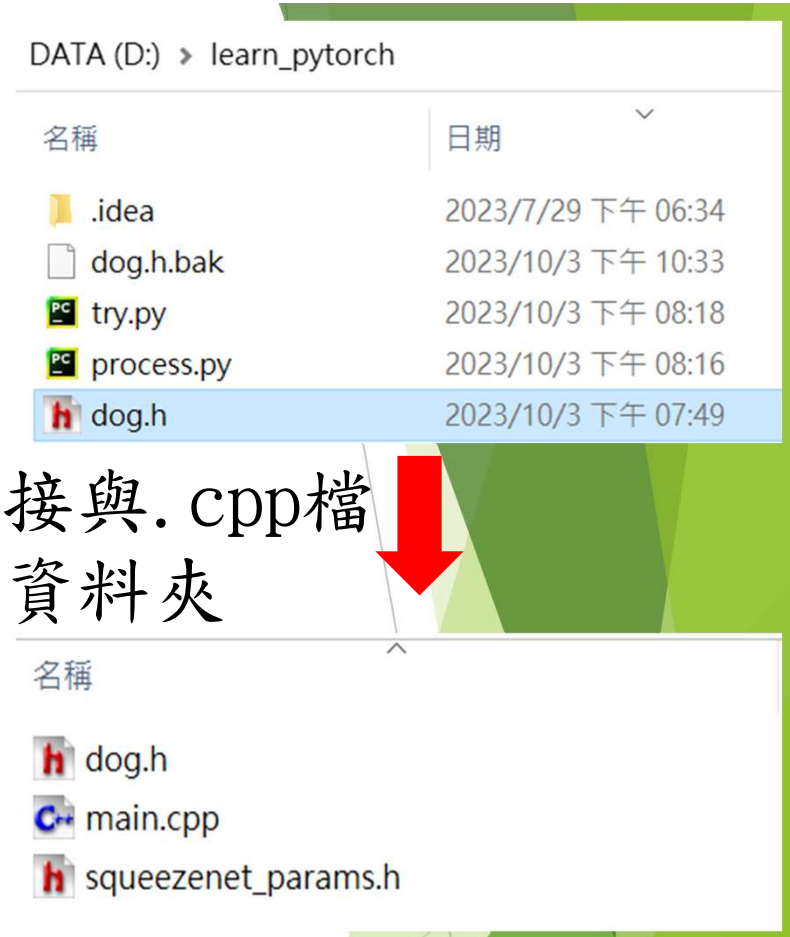

Transformed Image


Original Image


Resized Image (224x224)


Transformed Image

```python
# 将浮点数图像数据转换为一维数组
image_data = image.view(-1).numpy()

# 打印sample矩阵内的数值数量
print(f"Number of values in sample matrix: {len(image_data)}")

# 每行包含的浮点数数量
values_per_line = 8

# 创建一个.h文件并将数据写入其中
output_h_path = 'D:/learn_pytorch/dog.h'
with open(output_h_path, 'w') as h_file:
    h_file.write("float sample[] = {\n")
    for i, value in enumerate(image_data):
        h_file.write(f"{value:.6f}, ")
        if (i + 1) % values_per_line == 0:
            h_file.write("\n")
    h_file.write("};\n")
```



將.h檔直接與.cpp檔放入相同資料夾

```
C:\Users\ctchen.RTDOMAIN\AppData\Local\anaconda3\envs\d2l_pytorch\python.exe D:\learn_pytorch\change.py
Number of values in sample matrix: 150528

Process finished with exit code 0
```

dog.h

範例提供:

```
//235
float sample[] = {
0.930302, 0.913177, 0.896053, 0.878928, 0.861803, 0.844679, 0.844679, 0.861803,
0.878928, 0.896053, 0.896053, 0.913177, 0.930302, 0.913177, 0.913177, 0.896053,
0.878928, 0.878928, 0.861803, 0.861803, 0.861803, 0.896053, 0.930302, 0.964552,
0.998801, 1.03305, 1.05018, 1.08443, 1.10155, 1.1358, 1.15292, 1.17005,
```

自己產生(有中心裁剪):

```

float sample[] = {
0.930302, 0.913177, 0.896053, 0.878928, 0.861803, 0.844679, 0.844679, 0.861803,
0.878928, 0.896053, 0.896053, 0.913177, 0.930302, 0.913177, 0.913177, 0.896053,
0.878928, 0.878928, 0.861803, 0.861803, 0.861803, 0.896053, 0.930302, 0.964552,
0.998801, 1.033051, 1.050176, 1.084425, 1.101550, 1.135799, 1.152924, 1.170049,
```

自己產生(無中心裁剪):

```

float sample[] = {
0.622057, 0.604932, 0.587807, 0.587807, 0.587807, 0.604932, 0.604932, 0.587807,
0.587807, 0.570682, 0.570682, 0.553558, 0.587807, 0.604932, 0.639181, 0.656306,
0.741930, 0.827554, 0.878928, 0.947427, 1.015926, 1.067300, 1.118675, 1.152924,
1.170049, 1.170049, 1.152924, 1.084425, 0.981677, 0.878928, 0.810429, 0.741930,
```

重新編譯host文件
自行編寫Makefile:

```
23  # You must configure  INTELFPGAOCLSDKROOT to point the root directory of the Intel(R) FPGA SDK for OpenCL(TM)
24  # software installation.
25  # See http://www.altera.com/literature/hb/opencl-sdk/aocl_getting_started.pdf
26  # for more information on installing and configuring the Intel(R) FPGA SDK for OpenCL(TM).
27
28  ifeq ($(VERBOSE),1)
29  ECHO :=
30  else
31  ECHO := @
32  endif
33
34  # Where is the Intel(R) FPGA SDK for OpenCL(TM) software?
35  ifeq ($(wildcard $(INTELFPGAOCLSDKROOT)),)
36  $(error Set  INTELFPGAOCLSDKROOT to the root directory of the Intel(R) FPGA SDK for OpenCL(TM) software installation)
37  endif
38  ifeq ($(wildcard $(INTELFPGAOCLSDKROOT)/host/include/CL/opencl.h),)
39  $(error Set  INTELFPGAOCLSDKROOT to the root directory of the Intel(R) FPGA SDK for OpenCL(TM) software installation.)
```

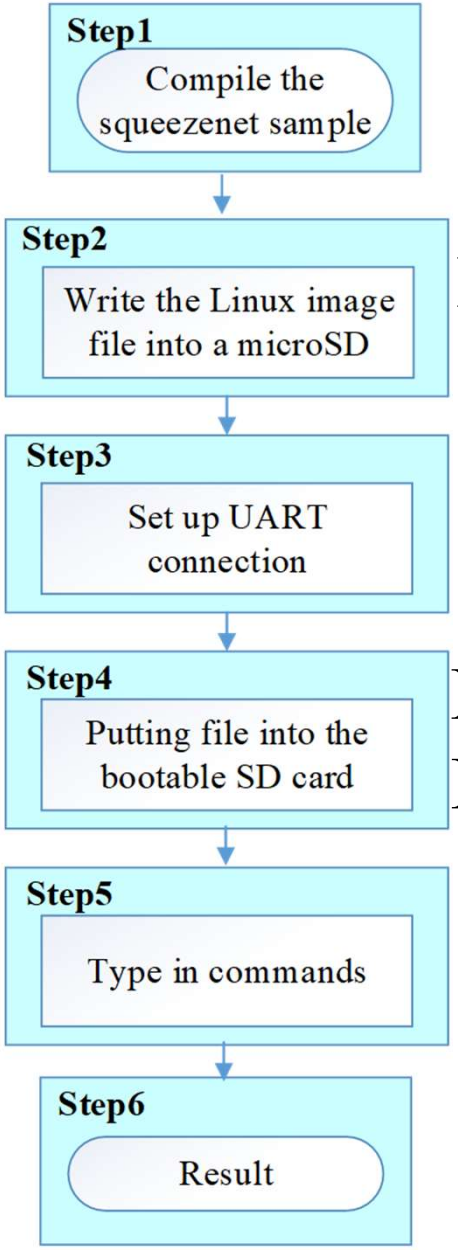
```
57  # Target
58  TARGET := squeezenet
59  TARGET_DIR := bin
60
61  # Directories
62  INC_DIRS := ../common/inc
63  LIB_DIRS :=
```

```
ctchen@R011432102 /cygdrive/C/intelFPGA/18.0/hld/board/terasic/de10_nano/examples/github
$ make

ctchen@R011432102 /cygdrive/C/intelFPGA/18.0/hld/board/terasic/de10_nano/examples/github
```

名稱

squeezenet

燒錄DE10-nano



此步驟已經完成

Download Win32DiskImager
de10_nano_opencl.img

Download PuTTY

RJ45 cable
Embedded_Command_Shell.bat

Running on DE10-Nano board

verification

測試圖片來源

Gallery of ImageNet Sample Images

The following gallery contains one sample image from each of the 1000 categories that ImageNet supports. The full dataset contains *many, many* images in each category. This gallery just gives you a taste of the full dataset.

tench (0)

`n01440764`

goldfish (1)

`n01443537`

great_white_shark (2)

`n01484850`

tiger_shark (3)

`n01491361`

狗

無中心裁剪：

```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 69.624 ms
block1 takes: 118.266 ms
block2 takes: 165.122 ms
block3 takes: 252.261 ms
classifier takes: 405.404 ms
total: 1010.677 ms

predicted label: n02106662 German shepherd, German shepherd dog, German police dog, alsatian
```

有中心裁剪：

```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 68.328 ms
block1 takes: 117.232 ms
block2 takes: 159.000 ms
block3 takes: 253.765 ms
classifier takes: 413.414 ms
total: 1011.739 ms

predicted label: n02106662 German shepherd, German shepherd dog, German police dog, alsatian
```

棒球選手

```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 68.183 ms
block1 takes: 116.921 ms
block2 takes: 155.911 ms
block3 takes: 255.701 ms
classifier takes: 401.729 ms
total: 998.445 ms

predicted label: n09835506 ballplayer, baseball player
```

枕頭

```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 68.022 ms
block1 takes: 117.321 ms
block2 takes: 159.449 ms
block3 takes: 242.022 ms
classifier takes: 403.482 ms
total: 990.297 ms

predicted label: n03938244 pillow
```

排球

```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 68.432 ms
block1 takes: 117.166 ms
block2 takes: 161.786 ms
block3 takes: 245.887 ms
classifier takes: 406.134 ms
total: 999.404 ms

predicted label: n04540053 volleyball
```

火把

```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 68.511 ms
block1 takes: 117.243 ms
block2 takes: 154.886 ms
block3 takes: 244.676 ms
classifier takes: 406.430 ms
total: 991.746 ms

predicted label: n03721384 marimba, xylophone
```

小提琴


```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 68.376 ms
block1 takes: 117.364 ms
block2 takes: 157.275 ms
block3 takes: 252.483 ms
classifier takes: 405.332 ms
total: 1000.830 ms

predicted label: n03794056 mousetrap
```

水壺

Soap Dispenser With Silver Pump

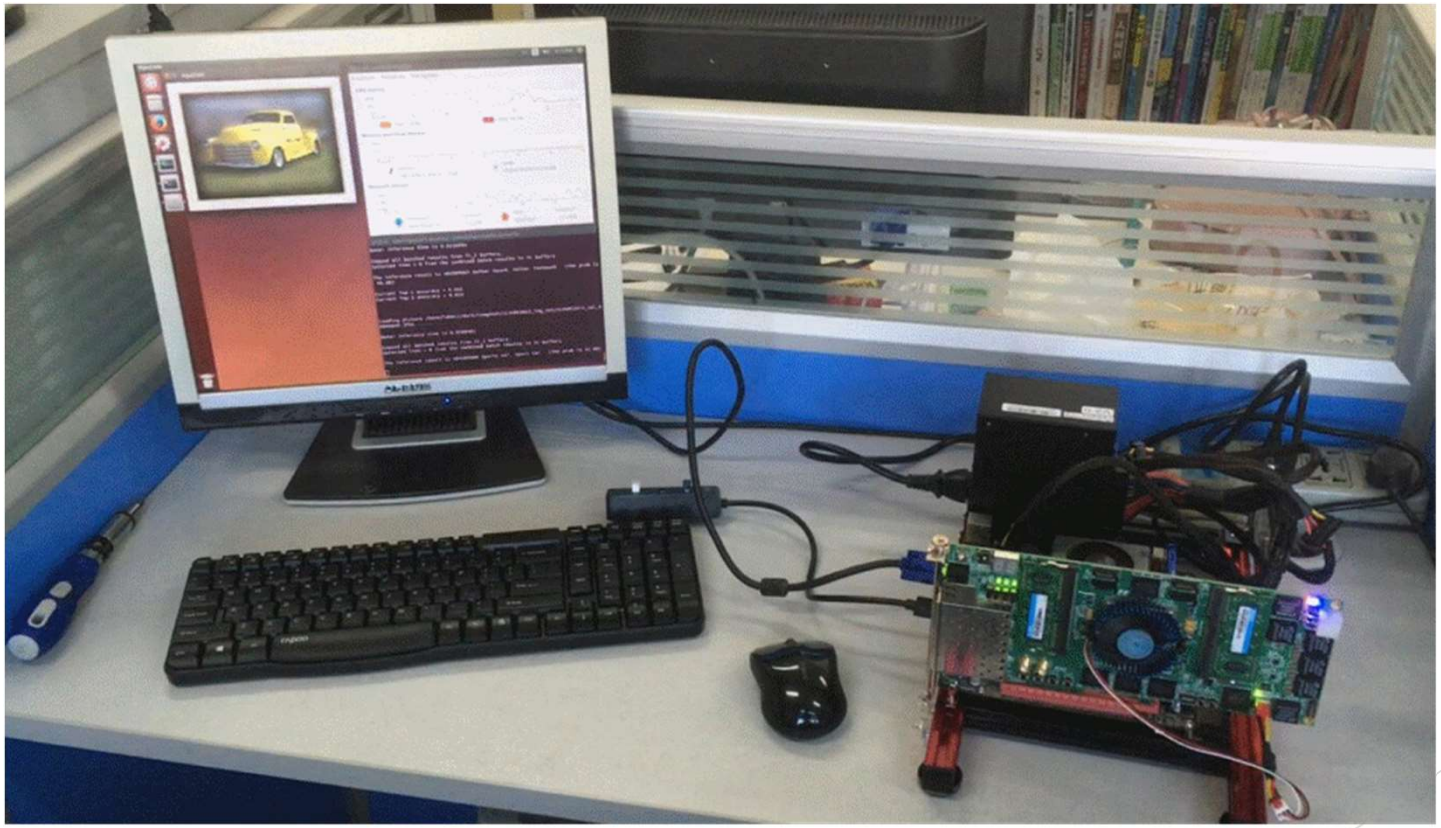
```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 68.341 ms
block1 takes: 117.064 ms
block2 takes: 158.319 ms
block3 takes: 250.766 ms
classifier takes: 403.081 ms
total: 997.571 ms

predicted label: n04254120 soap dispenser
```

人像


```
SqueezeNet on FPGA start:
kernel version 1.3

conv1 takes: 68.705 ms
block1 takes: 117.473 ms
block2 takes: 156.888 ms
block3 takes: 236.972 ms
classifier takes: 406.949 ms
total: 986.987 ms

predicted label: n03995372 power drill
```

下周研究方向

1. 實現連續辨識 2. 尋找相關應用



參考資料

1. Imagenet-sample
https://github.com/EliSchwartz/imagenet-sample-images#imagenet-sample-images
2. 4.1 Debugging OpenCL kernel with PyOpenCL
https://github.com/ErlcZ/Deploying_CNN_on_FPGA_using_OpenCL/blob/master/src/pyopencl/SqueezeNet.ipynb
3. TRANSFORMING AND AUGMENTING IMAGES
https://pytorch.org/vision/master/transforms.html
4. Pytorch transforms的常用操作
https://blog.csdn.net/Weary_PJ/article/details/113514906