Training object detection model

Training object detection model
- 1.Preparation before training
- 2.Download the basic model
- 3.Download image materials
- 4.started training
  - Configuration instructions for training parameters
- 5.Convert the model to ONnano
- 6.Import images to recognize
  - If you want to turn on the camera for recognition, you can enter the following command
- 7.recognition result

1.Preparation before training

Before starting this tutorial, Python must have the **torchCPU+GPU version installed, and the image of Yahboom is already installed**. If not installed, please install it yourself. You need to go online scientifically, and you can also read the Torch installation tutorial in this series of tutorials.

```
cd jetson-inference/build
./install-pytorch.sh
```

Because the basic model already has some basic information, retraining from scratch will greatly increase the training time. Therefore, this tutorial only explains retraining the basic model. If you want to retrain from scratch, please refer to the training tutorial in the previous chapter

2.Download the basic model

```
cd /home/jetson/jetson-inference/python/training/detection/ssd
wget https://nvidia.box.com/shared/static/djf5w54rjvpqocsiztzaandq1m3avr7c.pth -O
models/mobilenet-v1-ssd-mp-0_675.pth
pip3 install -v -r requirements.txt
```

If the model cannot be downloaded due to the inability to access the internet scientifically, you can find it in the/appendix/model file/fruit folder and transfer the model to the **/home/jetson/jetson-inference/python/training/detection/ssd/models**
The image of Yahboom does not require operation

3.Download image materials

```
cd /home/jetson/jetson-inference/python/training/detection/ssd
python3 open_images_downloader.py --class-names
"Apple,Orange,Banana,Strawberry,Grape,Pear,Pineapple,Watermelon" --
data=data/fruit
```

This command downloads all relevant images.
If your system space is insufficient or you don't want to save so many images, you can use the following methods to improve it
The image of YAHBOOM is to download all the relevant images

```
python3 open_images_downloader.py --stats-only --class-names
"Apple,Orange,Banana,Strawberry,Grape,Pear,Pineapple,Watermelon" --
data=data/fruit
```

In practice, in order to reduce training time (and disk space), you may want to keep the total number of images<10K. Although the more images you use, the more accurate your model will be.You can use to limit the amount of data downloaded--max-images Option or --max-annotations-per-class Option:

1. --stats-only：Download only images of the corresponding class, not unnecessary images
2. --max-images：Limit the total dataset to a specified number of images while maintaining a roughly identical image distribution for each class to the original dataset. If one class has more images than another, the proportion will remain roughly unchanged.
3. --max-annotations-per-classLimit each class to a specified number of bounding boxes. If a class has fewer available bounding boxes than that number, all of its data will be used - this is very useful if the distribution of data between classes is uneven.

For example, if you only want to use 2500 images for the fruit dataset, you can start the download program like this:

```
python3 open_images_downloader.py --max-images=2500 --class-names
"Apple,Orange,Banana,Strawberry,Grape,Pear,Pineapple,Watermelon" --
data=data/fruit
```

if --max-boxes option or --max-annotations-per-classnot set，By default, all available data will be downloaded - therefore, before doing so, be sure to use -- stats only.

4.started training

```
cd /home/jetson/jetson-inference/python/training/detection/ssd
python3 train_ssd.py --data=data/fruit --model-dir=models/fruit --batch-size=4 --
epochs=30
```

Note: If there is insufficient memory or if your process is "terminated" during training, please try installing swap and disabling desktop GUI.

```
#Disable Desktop GUI
sudo init 3 # stop the desktop
sudo init 5 # restart the desktop
```

Swap virtual memory (see link)：https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytor
ch-transfer-learning.md#mounting-swap

Configuration instructions for training parameters

configuration option	Default value	Description
--data	data/	Location of the dataset
--model-dir	models/	Output directory of trained model checkpoints
--resume	none	The path of the existing checkpoint to recover training from
--batch-size	4	Attempt to increase memory based on available memory
--epochs	30	Reaching 100 is desirable, but it will increase training time
--workers	2	Number of data loader threads (0=disable multithreading)

If you want to test your model before completing training in all eras, you can press Ctrl+C to terminate the training script and use -- resume=argument in the future
Our appendix/model file/fruit provides models trained for 30 cycles and models trained for 100 cycles. If you do not want to train, you can
Transfer the model to your path under nano by yourself/jetson-inference/python/training/detection/ssd/models/fruit， The image of yahboom does not require operation, as there are already 100 trained models in it

5.Convert the model to ONnano

Next, we need to convert the trained model from PyTorch to ONnano so that we can load it using TensorRT:

```
python3 onnano_export.py --model-dir=models/fruit
```

This will save a file named ssd mobilenet. onnano jetson-inference/python/training/detection/ssd/models/fruit/

6.Import images to recognize

If you have relevant images, you can upload them to nano for recognition. There are no images of 960 fruits that can be uploaded to nano under the appendix/model/fruit to /home/jetson/jetson-inference/python/training/detection/ssd/data， Namely (images. zip) and unzip。

```
export IMAGES=/home/jetson/jetson-
inference/python/training/detection/ssd/data/images
cd $IMAGES  && mkdir test && cd ../..
detectnet --model=models/fruit/ssd-mobilenet.onnano --
labels=models/fruit/labels.txt \
          --input-blob=input_0 --output-cvg=scores --output-bbox=boxes \
            "$IMAGES/fruit_*.jpg" $IMAGES/test/fruit_%i.jpg
```

The output results are in the
/home/jetson/jetson-inference/python/training/detection/ssd/data/test
Under file path
If you are using your own image and want to run the above command, you need to change the image name to fruit * Jpg, the image must be in jpg format, otherwise you can change the running command yourself. This tutorial will not elaborate on it anymore

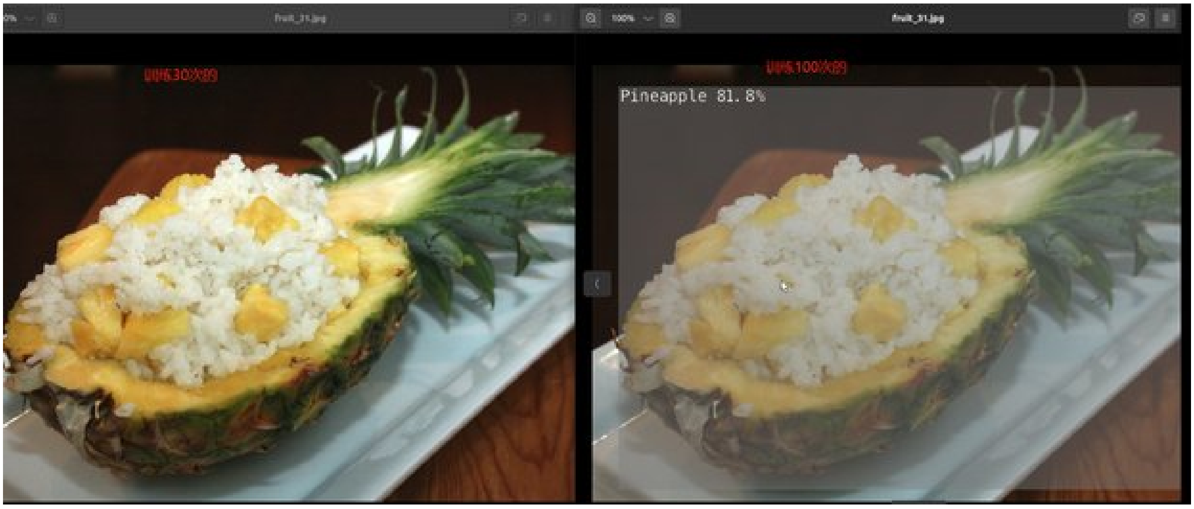
If you want to turn on the camera for recognition, you can enter the following command

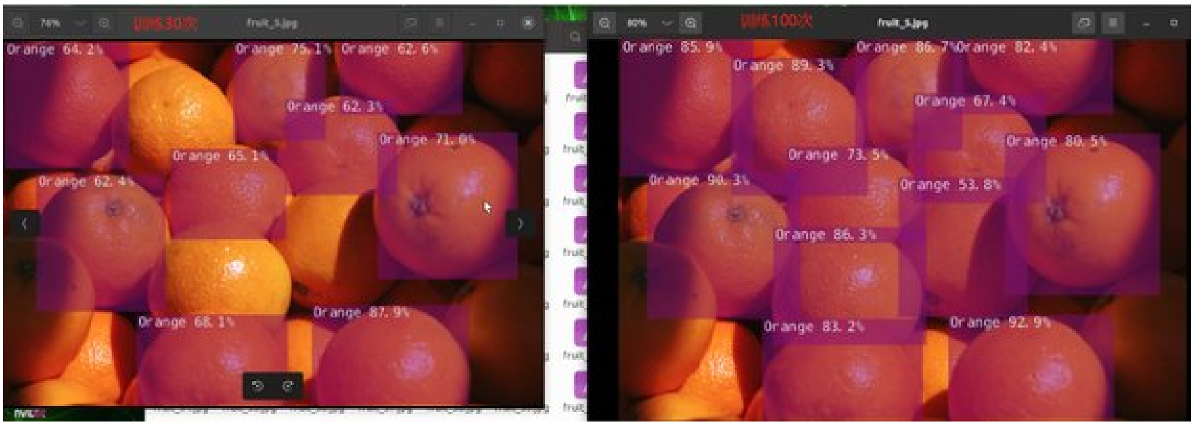
```
detectnet --model=models/fruit/ssd-mobilenet.onnano --
labels=models/fruit/labels.txt \
          --input-blob=input_0 --output-cvg=scores --output-bbox=boxes \
            csi://0     #csi

detectnet --model=models/fruit/ssd-mobilenet.onnano --
labels=models/fruit/labels.txt \
          --input-blob=input_0 --output-cvg=scores --output-bbox=boxes \
            v4l2:///dev/video0    #v4l2
```

For other video stream file opening methods, please refer to： https://github.com/dusty-nv/jetson-inference/blob/master/docs/aux-streaming.md

7.recognition result

There are recognition results in the image of YAHBOOM using models trained 30 times and models trained 100 times
30 results： /home/jetson/jetson-inference/python/training/detection/ssd/data/test_30
100 results:/home/jetson/jetson-inference/python/training/detection/ssd/data/test_100
Randomly select images for comparison in this tutorial
The results are shown in the figure：





It can be seen that the smaller the number of training sessions, the lower the accuracy, which may lead to the occurrence of unrecognizable images。