Train image classification models

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
```

2.Start retraining the cat and dog dataset

```
cd jetson-inference/python/training/classification/data
wget https://nvidia.box.com/shared/static/o577zd8yp3lmxf5zhm38svrbrv45am3y.gz -O
cat_dog.tar.gz
tar xvzf cat_dog.tar.gz
```

If the image cannot be downloaded successfully due to network issues, you can find cat in the "model/cat_dog" in the attachment_ Transfer dog.tar.gz to NX, yahboom can skip this step

3.Execute training commands

```
cd jetson-inference/python/training/classification
python3 train.py --model-dir=models/cat_dog data/cat_dog
```

Note: If the training is "terminated" due to insufficient memory, please try to exchange virtual memory and disable the 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/pytorch-transfer-learning.md#mounting-swap

4.Interpretation of information at the beginning of training

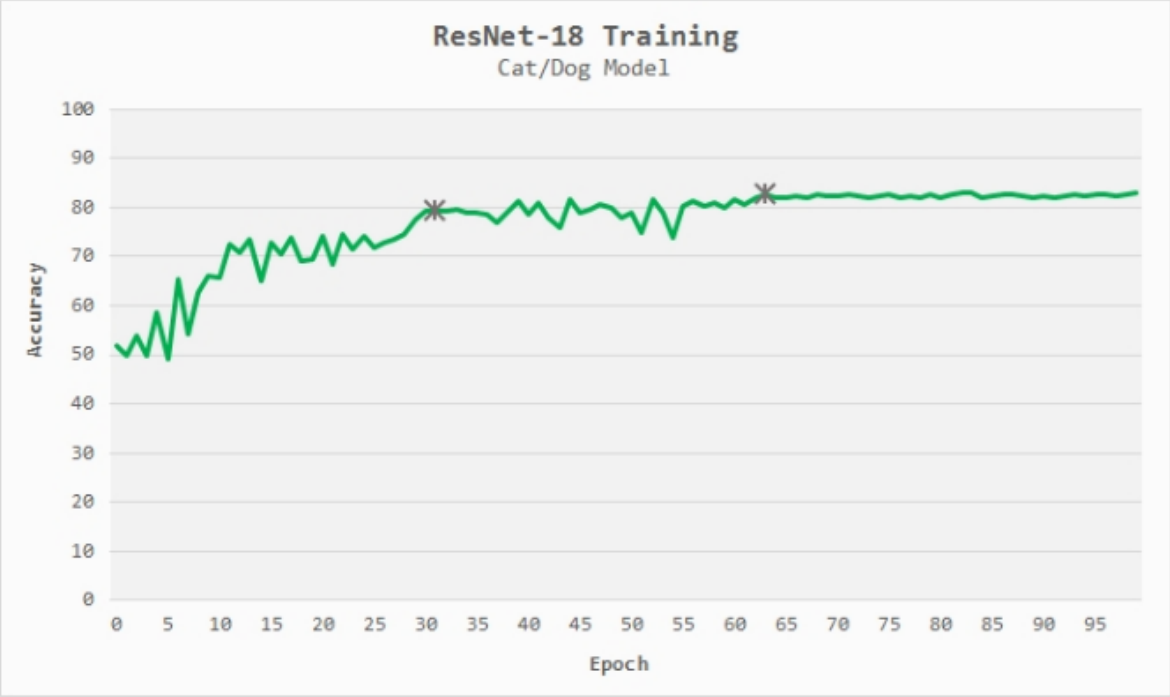
```
Use GPU: 0 for training
=> dataset classes:  2 ['cat', 'dog']
=> using pre-trained model 'resnet18'
=> reshaped ResNet fully-connected layer with: Linear(in_features=512,
out_features=2, bias=True)
Epoch: [0][  0/625] Time  0.932 ( 0.932)    Data  0.148 ( 0.148)    Loss
6.8126e-01 (6.8126e-01)    Acc@1  50.00 ( 50.00)   Acc@5 100.00 (100.00)
Epoch: [0][ 10/625] Time  0.085 ( 0.163)    Data  0.000 ( 0.019)    Loss
2.3263e+01 (2.1190e+01)    Acc@1  25.00 ( 55.68)   Acc@5 100.00 (100.00)
Epoch: [0][ 20/625] Time  0.079 ( 0.126)    Data  0.000 ( 0.013)    Loss
1.5674e+00 (1.8448e+01)    Acc@1  62.50 ( 52.38)   Acc@5 100.00 (100.00)
Epoch: [0][ 30/625] Time  0.127 ( 0.114)    Data  0.000 ( 0.011)    Loss
1.7583e+00 (1.5975e+01)    Acc@1  25.00 ( 52.02)   Acc@5 100.00 (100.00)
Epoch: [0][ 40/625] Time  0.118 ( 0.116)    Data  0.000 ( 0.010)    Loss
5.4494e+00 (1.2934e+01)
```

- Epoch: [0]:Represents the number of training cycles to 0: If no parameters are specified for the first one, the default is to only train 35 cycles. **You can use -- epochs=N to specify how many cycles to train, N=100, that is, to train 100 cycles**
- [N/625]:It is the current image batch in your era that trains images in small batches to improve performance. The default batch size is 8 images, and you can use -- batch=N to multiply the numbers in parentheses by the batch size (such as batch [100/625] ->images [800/5000])
  -Time:Processing time of the current image batch in seconds
  -Data:Disk loading time of the current image batch (in seconds)
  -Loss:Accumulated errors generated by the model (expected and predicted errors)
  If you want to train 10 times and batch process 6 images, run the command

```
python3 train.py --model-dir=models/cat_dog data/cat_dog --epochs=10 --batch=6
```

5.Explanation of model accuracy

On this dataset consisting of 5000 images, training ResNet-18 on Jetson orin NX takes approximately 2 minutes per period, or approximately 55 minutes to train the model to achieve 35 periods and 80% classification accuracy. The following figure is used to analyze the relationship between the training progress of epochs and the accuracy of the model:



In approximately the 30th period, the ResNet-18 model achieved an accuracy of 80%, and in the 65th period, it converged to an accuracy of 82.5%. With additional training time, you can further improve accuracy or try more complex models by increasing the size of the dataset.
By default, the training script is set to run for 35 periods, but if you don't want to wait that long to test your model, you can exit the training early and continue to the next step (you can choose to restart the training later from where you stopped)**Restore with the parameter '--resume'**
Our * * Appendix/Model File/cat_ Dog * * provides models for training 35 cycles and 100 cycles. If you don't want to train, you can transfer the model to NX yourself**your path/jetson-inference/python/training/classification/models/cat_dog**Just decompress it, but there is no need to manipulate the image of yahboom, as there are already 100 trained models in it

6.After training, convert the model into an ONNX model file

The model file of onnx is convenient for TensorRT to process

```
python3 onnx_export.py --model-dir=models/cat_dog
```

This will create a file named resnet18.onnx jetson-inference/python/training/classification/models/cat_dog/

7.Processing Images with TensorRT

Taking the command for yahboom mirroring as an example

```
cd /home/jetson/jetson-inference/python/training/classification
export NET=models/cat_dog
export DATASET=data/cat_dog
imagenet.py --model=$NET/resnet18.onnx --input_blob=input_0 --
output_blob=output_0 --labels=$DATASET/labels.txt $DATASET/test/cat/01.jpg
cat.jpg
```

The above command has been executed and the results can be observed in the path below. This command is used to process an image
path：/home/jetson/jetson-inference/python/training/classification

If you want to test all the images, you can enter the command

```
cd /home/jetson/jetson-inference/python/training/classification
export NET=models/cat_dog
export DATASET=data/cat_dog
imagenet --model=$NET/resnet18.onnx --input_blob=input_0 --output_blob=output_0 -
-labels=$DATASET/../labels.txt \
           $DATASET/test/dog ./data/cat_dog/test_output_dog
```

The output results are in the /home/jetson/jetson-inference/python/training/classification/data/cat_dog/test_output_dog

If you want to turn on the camera for detection

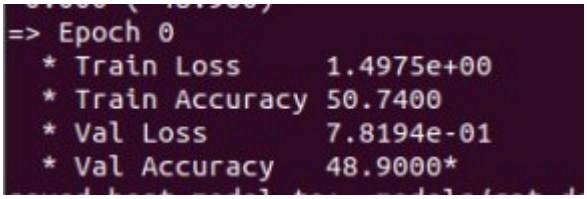
```
imagenet.py --model=$NET/resnet18.onnx --input_blob=input_0 --
output_blob=output_0 --labels=$DATASET/labels.txt csi://0    #CSI

imagenet.py --model=$NET/resnet18.onnx --input_blob=input_0 --
output_blob=output_0 --labels=$DATASET/labels.txt v4l2:///dev/video0    #V412
```

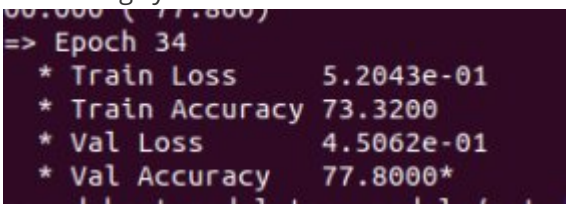
For other video stream file opening methods, please refer to：https://github.com/dusty-nv/jetson-inference/blob/master/docs/aux-streaming.md

8. Results of 1 training cycle, 35 training cycles, and 100 training cycles

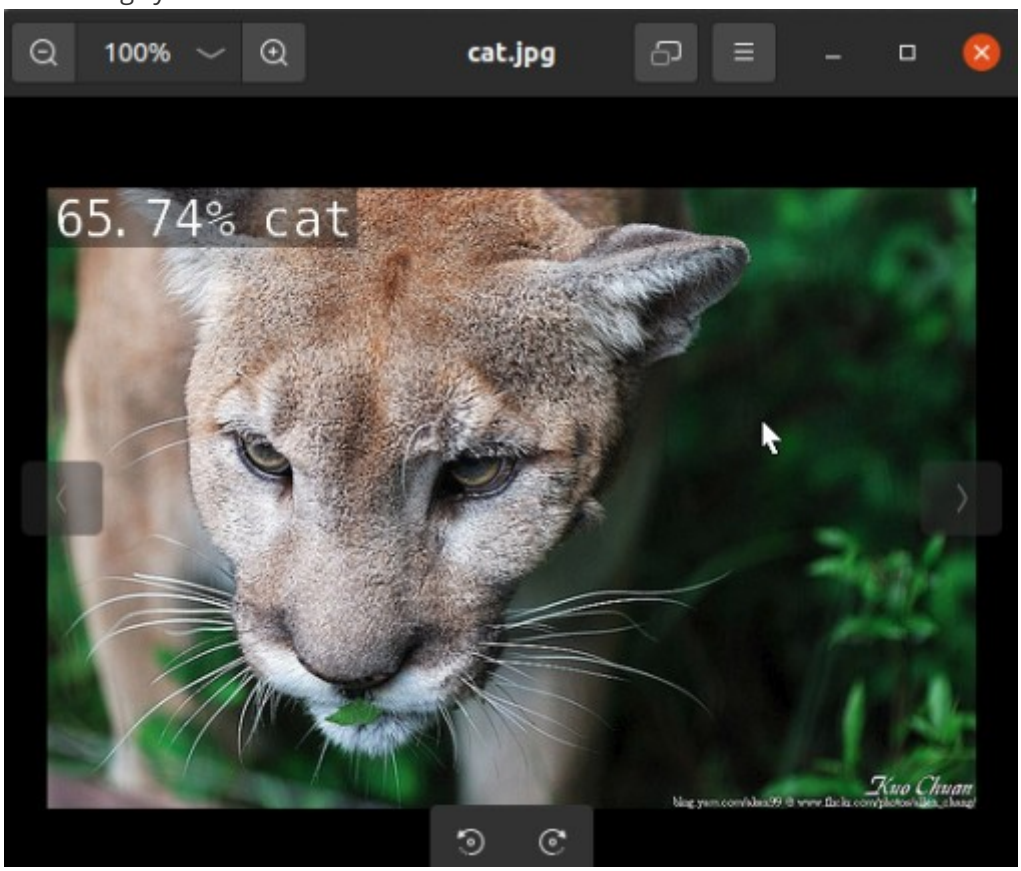
- 1 training cycle



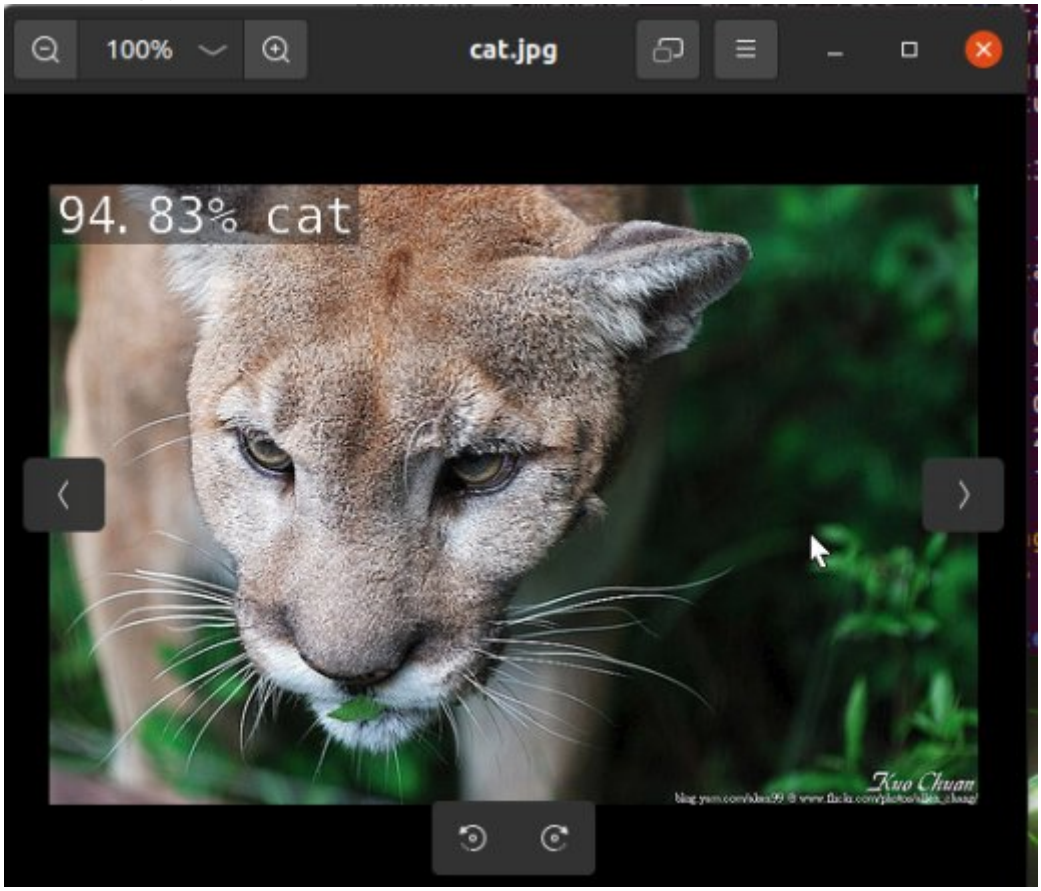
- 35 training cycles



- 35 training cycles



- 100 training cycles



It can be seen that the accuracy of the results from 100 training cycles is greatly improved compared to the results from 35 training cycles.

9.appendix

If you want to train yourself on different types of images

You can refer to the following link:

https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-collect.md