SUPPORTING INFORMATION

**Jetson Nano Tutorials**

**Workshop of computer vision for chemists: classify and detect microorganisms in optical microscopy images using machine learning**

Baosen Zhang,‡ Ariana Frkonja-Kuczin, ‡ Aliaksei Boika\*

Department of Chemistry, The University of Akron, Akron, OH 44325, USA

**Table of Contents Page**

Learning resources about Jetson Nano P1

Jetson Nano Introduction P2

The Linux command line basics P3

Installation of Jetson Inference P4

Example: Amoeba Detection P5

Bonus: Proteus Fowleri Detection P6

**Learning resources about Jetson Nano:**

* This is the original source of Jetson inference. They provide detailed tutorials of how to use the Jetson Nano or other NVIDIA product to train or do inference: [**https://github.com/dusty-nv/jetson-inference**](https://github.com/dusty-nv/jetson-inference)
* This is a Jetson Nano tutorial. It is friendly to the beginner: <https://www.youtube.com/playlist?list=PLGs0VKk2DiYxP-ElZ7-QXIERFFPkOuP4_&ab_channel=PaulMcWhorter>
* Ubuntu is one kind of Linux system and they provide command line tutorial for beginers: <https://ubuntu.com/tutorials/command-line-for-beginners#9-conclusion>
* Command line books: <http://linuxcommand.org/tlcl.php>

**Object Detection in Jetson Nano Introduction**

This document is the detailed tutorials for Level 3 (Object detection in Jetson Nano). At first, students need to learn Linux command line basics and know how to run the command line in Jetson Nano device. Then if students get a new Jetson Nano device, they will take time to install Jetson Inference project on it. After building up the project, students need to download the project in github(<https://github.com/BaosenZ/amoeba-detection> ) and save the Jetson Nano dataset to correct folder, and start to train by command. After training, we can evaluate the performance and run inference for the drop or test dataset by command. The results are shown in the Figure S6-10. For the assignment and solution, students follow the same procedure to difference two types of amoeba with the command line with the dataset of Proteus and Fowleri that we provided.

**The Linux command line basics**

One of the early operating systems is called Unix. At that time, there was no mouse, no fancy graphics, not even any choice of color, but programmer produce text as output and accept text as an input via Unix mainframe. Linux is a sort-of-descendent of Unix. The core part of Linux is designed to behave similarly to a Unix System. The Linux command line, like Unix mainframe, is a text interface to your computer. Although running Windows graphical programs is more common in the recent days, learning text interface of Linux command line is still meaningful, since it can let us understand more underlying logic of the computer. This will also increase our efficiency dealing with computer programs after being familiar with Linux command line.

The command can finish tasks like changing directories (cd), listing the contents (ls), moving files (mv), and so on. Here, we provide the common used command line as a cheat sheet for students to have a quick reference (Table 1). The Figure 1 is the example of Linux command line terminal. More Linux command line learning resources can check the video in YouTube (<https://www.youtube.com/watch?v=MfpvdC-QrgY&list=PLGs0VKk2DiYxP-ElZ7-QXIERFFPkOuP4_&index=2&t=2117s&ab_channel=PaulMcWhorter>.), the Ubuntu command line tutorials (<https://ubuntu.com/tutorials/command-line-for-beginners#9-conclusion>.), and Linux command line books(<http://linuxcommand.org/tlcl.php>.).

Table 1. Common used Linux command line:

|  |  |
| --- | --- |
| Linux command line | Explanation |
| $ pwd | Show current directory |
| $ mkdir dir | Make directory called dir |
| $ rm –r dir | Delete directory called dir |
| $ cd dir | Change directory to dir |
| $ cd .. | Go up a directory |
| $ ls | List files |
| $ touch file1 | Create a file called file1 |
| $ file file1 | Get type of file1 |
| $ cp file1 file2 | Copy file1 to file2 |
| $ mv file1 file2 | Move file1 to file2 |
| $ rm file1 | Delete file1 |
| $ ctrl-c | Stop current command |
| $ clear | Clear all command line |
| Tab | Fill out the command line automatically |

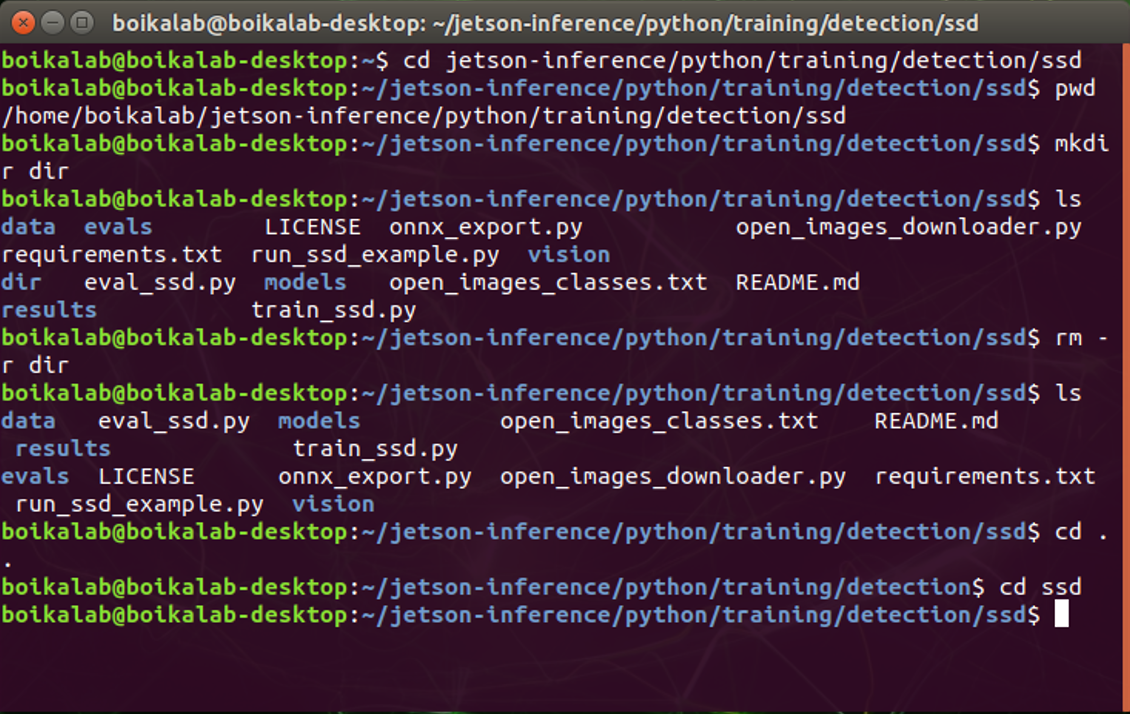


Figure 1. Example of Linux command line.

**Installation of Jetson Inference project from source**

Installing the Jetson Inference project from the source is a start to operate Jetson Nano device and can work as a test for students for Linux command line. The Jetson inference project includes some deep learning networks for image classification, object detection, and semantic segmentation. Various pre-trained deep learning models are automatically downloaded to get you up and running quickly. We can also train new models with the pre-trained networks to get our own custom model. The Box 1 provide the condensed form of the commands to install the project. More details of installation process can be referred to the source (<https://github.com/dusty-nv/jetson-inference/blob/master/docs/building-repo-2.md> ) and the video (<https://www.youtube.com/watch?v=5rbOsKCZ-VU&list=PLGs0VKk2DiYxP-ElZ7-QXIERFFPkOuP4_&index=49&ab_channel=PaulMcWhorter> ).

Box 1. The condensed form of the commands to build/install the project directly on your Jeston Nano.

$ sudo apt-get update

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

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

$ cd jetson-inference

$ mkdir build

$ cd build

$ cmake ../

$ make -j$(nproc)

$ sudo make install

$ sudo ldconfig

**Example: Amoeba Detection**

The example for level 3 is amoeba detection. Students can go to the Github website (<https://github.com/BaosenZ/amoeba-detection> ) to download the zip of folders, then drag the dataset to the correct directory. Before training, we need to do mounting swap to free extra memory (<https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-transfer-learning.md#mounting-swap> ). After installation of the Jetson Inference project, students can run the command line in Box 2 to train and evaluate the model. For the train\_ssd.py, it involves many arguments, type $ python3 train\_ssd.py help to see all arguments. Box 3 is the python code to run the inference.

Box 2. The condensed form of the commands to train and evaluate the model.

$ cd jetson-inference/python/training/detection/ssd

# Start to train the model

$ python3 train\_ssd.py --data=data/amoebaDataset/trainingDataset --model-dir=models/amoebaModel\_mb2 --dataset-type=voc --net=mb2-ssd-lite --pretrained-ssd=models/mb2-ssd-lite-mp-0\_686.pth --epochs=60

# Convert the model to ONNX

$ python3 onnx\_export.py --model-dir=models/amoebaModel\_mb2 --net=mb2-ssd-lite

# Evaluate the model

$ python3 eval\_ssd.py –trained\_model=models/amoebaModel\_mb2/<.pth> --net=mb2-ssd-lite --eval\_dir=evals\_amoebaEval --label\_file=models/amoebaModel/label.txt --dataset=data/amoebaDataset/testDataset

Box 3. Python version of the inference.

import jetson.inference

import jetson.utils

import cv2

import matplotlib.pyplot as plt

import numpy as np

net = jetson.inference.detectNet('ssd-mobilenet',['--model=/home/boikalab/jetson-inference/python/training/detection/ssd/models/amoebaModel\_mb2/mb2-ssd-lite.onnx','--input\_blob=input\_0','--output\_cvg=scores','--output\_bbox=boxes','--labels=/home/boikalab/jetson-inference/python/training/detection/ssd/models/amoebaModel\_mb2/labels.txt', threshold=0.5])

for i in range(1,21):

# Put the original images to ‘c1’ folder.

img = cv2.imread("./c1/" + str(i) + ".jpg")

img = jetson.utils.cudaFromNumpy(img)

detections=net.Detect(img)

img = np.ascontiguousarray(img)

ImgName = str(i) + ".jpg"

cv2.imwrite(ImgName,img)

**Bonus: Proteus Fowleri Detection**

Students can go to the Github website (<https://github.com/BaosenZ/amoeba-detection> ) to download the zip of folders, then drag the dataset to the correct directory. Based on the command line and python code in Box 2, 3 in the Example, students can train, evaluate and do inference for the A. proteus and N.fowleri amoeba dataset.

Box 2. The condensed form of the commands to train and evaluate the model.

$ cd jetson-inference/python/training/detection/ssd

# Start to train the model

$ python3 train\_ssd.py --data=data/amoebaProFowDataset/trainingDataset --model-dir=models/amoebaProFowModel\_mb2 --dataset-type=voc --net=mb2-ssd-lite --pretrained-ssd=models/mb2-ssd-lite-mp-0\_686.pth --epochs=100

# Convert the model to ONNX

$ python3 onnx\_export.py --model-dir=models/amoebaProFowModel\_mb2 --net=mb2-ssd-lite

# Evaluate the model

$ python3 eval\_ssd.py –trained\_model=models/amoebaProFowModel\_mb2/<.pth> --net=mb2-ssd-lite --eval\_dir=evals\_amoebaProFowEval --label\_file=models/amoebaProFowModel/label.txt --dataset=data/amoebaProFowDataset/testDataset

Box 3. Python version of the inference.

import jetson.inference

import jetson.utils

import cv2

import matplotlib.pyplot as plt

import numpy as np

net = jetson.inference.detectNet('ssd-mobilenet-v2',['--model=/home/boikalab/jetson-inference/python/training/detection/ssd/models/amoebaProFowModel/mb2-ssd-lite.onnx','--input\_blob=input\_0','--output\_cvg=scores','--output\_bbox=boxes','--labels=/home/boikalab/jetson-inference/python/training/detection/ssd/models/amoebaProFowModel/labels.txt','--threshold=0.2'])

for i in range(63,73):

img = cv2.imread("./testDataset/" + str(i) + ".jpg")

img = jetson.utils.cudaFromNumpy(img)

detections=net.Detect(img)

img = np.ascontiguousarray(img)

ImgName = str(i) + ".jpg"

cv2.imwrite(ImgName,img)