**Advanced AI short course: Jetson Nano**

**Running the Docker Container**

1. Google “dusty-nv jetson inference”

2. Click the first search result “Hello AI World guide to deploying deep-learning … – GitHub”

3. Click “Running the Docker Container” under Hello AI world – System Setup

4. Open Terminal on Desktop

5. Download jeston-inference and docker on host by command:

**a. git clone --recursive --depth=1 https://github.com/dusty-nv/jetson-inference**

**b. cd jetson-inference**

**c. docker/run.sh**

6. Wait until download completed (around 5 minutes). After completion, the terminal is connected to the container

In summary, on the container, you see the line beginning with “root@emsd-desktop”.

On the host, you see the line beginning with “emsd@emsd-desktop”.

7. Check jetson-inference folder on GUI under Home directory

8. Talk about docker and container

9. Mounted Data Volumes - the following paths automatically get mounted from your host device into the container:

a. jetson-inference/data (stores the network models, serialized TensorRT engines, and test images)

b. jetson-inference/python/training/classification/data (stores classification training datasets)

c. jetson-inference/python/training/classification/models (stores classification models trained by PyTorch)

d. jetson-inference/python/training/detection/ssd/data (stores detection training datasets)

e. jetson-inference/python/training/detection/ssd/models (stores detection models trained by PyTorch)

10. Check USB camera on container by command:

a. **cd /jetson-inference/build/aarch64/bin**

b. **./video-viewer /dev/video0**

(Before running this command, make sure your USB camera is well connected to the Jetson Nano. After running this command, a window jumps out showing the view of the USB camera.)

11. More reference can be found by “Camera Streaming and Multimedia” under Hello AI World- Appendix

**Object Detection - Running the Live Camera Detection Demo**

1. The paths jetson-inference/data automatically get mounted from your host device into the container

2. Check images folder and networks folder on container by command:

a. **cd /jetson-inference/build/aarch64/bin**

b. **ls**

3. Check testing image on GUI under home/jetson-inference/data/images/peds\_0.jpg

4. Run detection model on container by command:

a. **cd /jetson-inference/build/aarch64/bin**

b. **./detectnet.py images/peds\_0.jpg images/test/peds\_0.jpg**

(around 5 minutes)

5. Downloading ssd object detection model and perform TensorRT

6. Check result image on GUI under home/jetson-inference/data/images/test/peds\_0.jpg

7. Check the ssd model on GUI under home/jetson-inference/data/networks

8. Check the python script on GUI under

home/jetson-inference/python/examples/detectnet.py

9. Run detection model in real time on container by command:

a. **cd /jetson-inference/build/aarch64/bin**

b. **./detectnet.py /dev/video0**

(around 4 minutes)

**Object Detection - Coding Your Own Object Detection Program**

1. Create your own python script on host by command:

a. **cd ~**

**b. mkdir my-detection**

**c. cd my-detection**

**d. touch my-detection.py**

**e. cd ~/jetson-inference**

**f. docker/run.sh --volume ~/my-detection:/my-detection**

2. Check the python script on container by command:

a. **ls /my-detection**

**b. cd /my-detection**

3. Copy the code on GUI from home/jetson-inference/python/example/my-detection.py to home/my-detection/my-detection.py

4. Modify the python script on GUI under home/my-detection/my-detection.py

a. camera = videoSource("csi://0") => camera = videoSource("/dev/video0 ")

b. display = videoOutput("display://0") => display = videoOutput()

c. add the below code after detections = net.Detect(img)

|  |
| --- |
| if(len(detections)>0):  for detection in detections:  ClassLabel = net.GetClassLabel(detection.ClassID)  Center = detection.Center  print("Class:{:s}, Center:{} ".format(ClassLabel, Center)) |

5. Reference of detectNet : https://rawgit.com/dusty-nv/jetson-inference/master/docs/html/python/jetson.inference.html#detectNet

6. Check and run the python script on container by command:

a. **cd /my-detection**

**b. cat my-detection.py**

**c. python3 my-detection.py**

**Transfer Learning with PyTorch - Collecting your own Detection Datasets**

1. Check memory on host by command

a. **free -m**

2. Mounting Swap on host by command

a. **sudo systemctl disable nvzramconfig**

**b. sudo fallocate -l 4G /mnt/4GB.swap**

**c. sudo mkswap /mnt/4GB.swap**

**d. sudo swapon /mnt/4GB.swap**

**e. echo '/mnt/4GB.swap none swap sw 0 0' | sudo tee -a /etc/fstab**

**f. cat /etc/fstab**

3. Restart jeston-nano

4. Check memory on host by command

a. **free -m**

5. create data folder on GUI under Home/jetson-inference/python/training/detection/ssd/data

a. Right click

b. Create a New Folder called custom

c. Go into custom folder

d. Right click

e. Open in Terminal

f. Input command in terminal: **touch labels.txt**

g. Open lables.txt and type jeston-nano

6. Collecting custom datasets on container by command:

a. **cd /jetson-inference/python/training/detection/ssd**

b. **camera-capture /dev/video0**

c. For Data Capture Control:

i. Dataset Type: Detection

ii. Dataset Path: /jetson-inference/python/training/detection/ssd/data/custom

iii. Class Labels: /jetson-inference/python/training/detection/ssd/data/custom/labels.txt

iv. Current set: train

d. Capture image by Freeze/Edit(space) button

e. Make bounding box on image

f. Release the camera by Freeze/Edit(space) button

g. Check Annotation folder, ImageSets folder , and JPEGImage folder on GUI under /jetson-inference/python/training/detection/ssd/data/custom/

h. Capture about 15 images in different distance and angle

i. Change Current set from train to test

j. Capture about 5 images in different distance and angle

7. Train the model on container by command:

a. **cd /jetson-inference/python/training/detection/ssd**

**b.** **python3 train\_ssd.py --dataset-type=voc --data=data/custom --model-dir=models/custom --batch-size=2 --workers=1 --epochs=100**

8. Converting the Model to ONNX on container by command:

a. **python3 onnx\_export.py --model-dir=models/custom**

9. Run the custom model on container by command:

a. **detectnet --model=models/custom/ssd-mobilenet.onnx --labels=models/custom/labels.txt --input-blob=input\_0 --output-cvg=scores --output-bbox=boxes /dev/video0**